

AD-A172 929

ARO 22848.2-ELS
②

JC-2039-A

THE EFFECTIVENESS OF RANDOM MFSK FREQUENCY-HOPPING ECCM RADIOS AGAINST WORST-CASE PARTIAL-BAND NOISE JAMMING

FINAL REPORT

AUGUST 1986

PREPARED FOR
U. S. ARMY RESEARCH OFFICE

CONTRACT
DAAG29-85-C-0021

The views, opinions, and/or findings contained in this report
are those of the authors and should not be construed as an
official Department of the Army position, policy, or decision,
unless so designated by other documentation.

REPRODUCED FROM
BEST AVAILABLE COPY



DTIC
ELECTED
OCT 08 1986
[Handwritten signature]
S D

J. S. LEE ASSOCIATES, INC.
2001 JEFFERSON DAVIS HIGHWAY, SUITE 601
ARLINGTON, VIRGINIA 22202

Approved for public release; distribution unlimited.

393892 ptm 86 10 0 089

**THE EFFECTIVENESS OF
RANDOM MFSK
FREQUENCY-HOPPING ECCM RADIOS
AGAINST WORST-CASE
PARTIAL-BAND NOISE JAMMING**

FINAL REPORT

AUGUST 1986

**PREPARED FOR
U.S. ARMY RESEARCH OFFICE**

**CONTRACT
DAAG29-85-C-0021**

The views, opinions, and/or findings contained in this report are those of the authors and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

**J. S. LEE ASSOCIATES, INC.
2001 JEFFERSON DAVIS HIGHWAY, SUITE 601
ARLINGTON, VIRGINIA 22202**

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER <i>ARO 22848.2-EL-5</i>	2. GOVT ACCESSION NO. <i>N/A</i>	3. RECIPIENT'S CATALOG NUMBER <i>N/A</i>
4. TITLE (and Subtitle) <i>The Effectiveness of Random MFSK Frequency-Hopping ECCM Radios Against Worst-Case Partial-Band Noise Jamming</i>		5. TYPE OF REPORT & PERIOD COVERED <i>Final Report 1 Jul 1985-31 July 1986</i>
7. AUTHOR(s) <i>L.E. Miller, R.H. French, L.J. Cowles, and J.S. Lee</i>		6. PERFORMING ORG. REPORT NUMBER <i>JC-2039-A</i>
9. PERFORMING ORGANIZATION NAME AND ADDRESS <i>J.S. Lee Associates, Inc. Suite 601, 2001 Jefferson Davis Highway Arlington, Virginia 22202</i>		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS <i>N/A</i>
11. CONTROLLING OFFICE NAME AND ADDRESS <i>U. S. Army Research Office Post Office Box 12211 Research Triangle Park NC 27709</i>		12. REPORT DATE <i>August 1986</i>
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		13. NUMBER OF PAGES <i>528</i>
		15. SECURITY CLASS. (of this report) <i>Unclassified</i>
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) <i>Approved for public release; distribution unlimited.</i>		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) <i>NA</i>		
18. SUPPLEMENTARY NOTES <i>The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.</i>		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) <i>Frequency-shift-keying (FSK), Frequency-hopping (FH), Jamming, Probability of error, Diversity, ECCM</i>		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) <i>The system analyzed for bit error rate (BER) performance is frequency-hopped (FH) multiple frequency-shift-keying (MFSK) wherein the M symbol frequency assignments are independently and randomly chosen on each hop, and the information transmission is repeated L times (L-hop diversity) for a soft symbol decision. This FH/RMFSK system is intended to counter systematic follow-on jamming. The reported analysis is of the system's BER in worst case partial-band noise jamming (WCPBNJ). To be effective, the receiver must employ nonlinear combining of the L hops; several hop weighting schemes are evaluated with different assumptions about</i>		

Block 18 (cont.)

Further supplemental information:

- a. Participating scientific personnel: authors (see list in Block 4) plus H.M. Kwon
- b. Degrees earned while employed on project: none
- c. Publications connected with this work (to date):
L.E. Miller, R.H. French, and J.S. Lee, "Performance of Frequency-Hopped Random MFSK in Follow-on and Partial-band Noise Jamming," to be presented at IEEE GLOBECOM '86 Conference and included in the proceedings of that conference.

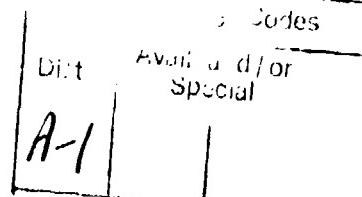
Block 20 (cont.)

available real-time information on relative powers of signal, noise, and jamming. These include adaptive gain control, clipping, hard-decision, and self-normalizing (nonparametric) schemes. It is shown that a simple, self-normalizing receiver, using no jamming state information or measurements, can perform nearly as well as one using a priori values of received noise-plus-jamming powers for adaptive gain control. It is also demonstrated that a hard-decision receiver (majority logic decoding of the L repetitions) achieves an ECCM effect and is viable if the SNR is high. Although the BER varies with jammer power in much the same way as for conventional FH/MFSK (given the parameters M, L, and the unjammed SNR), including a diversity gain for high SNR, FH/RMFSK in general is more vulnerable to WCPBNJ for M greater than 2. Therefore, it is concluded that implementation of effective diversity schemes is feasible, and that for a binary system the additional complexity of random hopping can be assessed to the additional protection gained against follow-on jamming..

J. S. LEE ASSOCIATES, INC.

TABLE OF CONTENTS

	<u>PAGE</u>
1.0 INTRODUCTION.	1-1
1.1 BACKGROUND	1-1
1.1.1 Jamming Strategy Against Frequency-Hopping Radios	1-3
1.1.2 ECCM Waveforms	1-4
1.1.3 Motivation for the Proposed Random Hopping	1-7
1.1.4 Rationale for the Exact Analysis of FH/MFSK.	1-9
1.1.5 Extension of Uncoded Error Analysis to Coded Performance	1-11
1.2 ECCM PROCESSING	1-15
1.2.1 Examples of Receiver Effectiveness Computations for FH/MFSK	1-16
1.2.2 Impact of Random FH/MFSK (FH/RMFSK) on Analysis	1-25
1.3 SUMMARY OF REPORT	
1.3.1 General Description of Work and Approach	1-27
1.3.1.1 Receiver models studied	1-27
1.3.1.2 Jamming model and measure of effectiveness	1-31
1.3.1.3 Organization of report	1-34
1.3.2 Summary of Findings	1-35
2.0 PRELIMINARY ANALYSIS	2-1
2.1 CONDITIONAL PROBABILITY OF ERROR	2-1
2.1.1 Assumed Signals, Noise and Jamming	2-3
2.1.2 Conditional Error Probability Formulation	2-5



J. S. LEE ASSOCIATES, INC.

TABLE OF CONTENTS (Continued)

	<u>PAGE</u>
2.2 ENUMERATION OF DISTINGUISHABLE JAMMING EVENTS	2-7
2.2.1 Definition of Distinguishable Jamming Events . . .	2-7
2.2.2 Smallest Set of Distinguishable Jamming Events . . .	2-8
2.2.3 Jamming Event Probabilities, Single Hop	2-11
2.2.4 Jamming Event Probabilities for Multiple Hops, Characteristic Function Method	2-13
2.2.5 Jamming Event Probabilities For Multiple Hops Convolution Method	2-16
2.3 TOTAL PROBABILITY OF ERROR	2-20
2.4 FH/RMFSK HARD DECISION ANALYSIS	2-21
2.4.1 Formulation of Error Probability	2-21
2.4.2 Explicit Form of Error Probability	2-23
2.4.2.1 Special case: L=2	2-25
2.4.2.2 Special case: L=3	2-25
2.4.2.3 Special case: L=4	2-26
2.4.2.4 Special case: M=2	2-26
3.0 FH/RMFSK PERFORMANCE USING SQUARE-LAW LINEAR COMBINING RECEIVER	3-1
3.1 ERROR PROBABILITY ANALYSIS	3-1
3.1.1 Distribution of the Decision Statistics	3-2
3.1.2 Formulation of the Conditional Error Probability	3-4
3.1.3 Powers of Non-Signal Channel Probabilities . . .	3-5
3.1.4 Expectation Over Signal Channel PDF	3-11
3.1.5 Special case: L=1 and M=2	3-18

J. S. LEE ASSOCIATES, INC.

TABLE OF CONTENTS (Continued)

	<u>PAGE</u>
3.2 NUMERICAL RESULTS	3-19
3.2.1 Soft-Decision Receiver	3-19
3.2.2 Hard-Decision Receiver	3-33
4.0 FH/RMFSK PERFORMANCE USING SQUARE-LAW AGC RECEIVER	4-1
4.1 POSSIBLE AGC WEIGHTING SCHEMES FOR FH/RMFSK	4-2
4.1.1 Average AGC Scheme	4-2
4.1.2 Individual Channel AGC Scheme	4-4
4.1.3 Any-Channel-Jammed AGC Scheme	4-5
4.2 ANALYSIS OF FH/RMFSK PERFORMANCE USING INDIVIDUAL CHANNEL AGC SCHEME	4-7
4.2.1 Conditional Probability of Error	4-7
4.2.2 Total Error Probability	4-9
4.3 ANALYSIS OF FH/RMFSK PERFORMANCE USING ANY-CHANNEL-JAMMED AGC SCHEME	4-9
4.3.1 Conditional Probability of Error	4-9
4.3.1.1 Formulations of nonsignal channel probabilities	4-11
4.3.1.2 Formulation of symbol error probability in terms of previous results	4-12
4.3.2 Enumeration and Probabilities for Jamming Events	4-13
4.4 NUMERICAL RESULTS	4-18
4.4.1 Numerical Results for Individual-Channel AGC Receiver	4-18
4.4.2 Numerical Results for Any-Channel-Jammed AGC (ACJ-AGC) Receiver	4-28

J. S. LEE ASSOCIATES, INC.

TABLE OF CONTENTS (Continued)

	<u>PAGE</u>
5.0 FH/RMFSK PERFORMANCE USING CLIPPER RECEIVER	5-1
5.1 DISTRIBUTIONS OF THE DECISION VARIABLES	5-1
5.1.1 General Form of the pdf	5-1
5.1.2 Non-Signal Channel pdf	5-10
5.1.3 Signal Channel pdf	5-13
5.2 ERROR PROBABILITY FORMULATION	5-15
5.2.1 Conditional Probability of Error	5-15
5.2.1.1 Special case: $L=1$ (one hop/symbol)	5-20
5.2.1.2 Special case: $\underline{e}=0$ (no jamming)	5-22
5.2.2 Total Probability of Error	5-24
5.2.2.1 Choice of clipping threshold	5-25
5.2.2.2 Total error for $L=1$	5-26
5.3 NUMERICAL RESULTS	5-27
5.3.1 The Optimum Threshold Setting	5-28
5.3.2 Probability of Bit Error	5-30
6.0 FH/RMFSK PERFORMANCE USING SQUARE-LAW SELF-NORMALIZATION RECEIVER	6-1
6.1 THE SELF-NORMALIZATION SCHEME	6-1
6.1.1 Single-hop Distribution of Decision Variables	6-2
6.1.2 Alternate Forms	6-5
6.1.2.1 Special case: $M=2$ (binary)	6-6
6.1.2.2 Special case: no jamming	6-6

J. S. LEE ASSOCIATES, INC.

TABLE OF CONTENTS (Continued)

	<u>PAGE</u>
6.1.3 Conditional pdf's for M=2	6-7
6.1.3.1 Single hop/bit case (L=1)	6-7
6.1.3.2 Two hops/bit case (L=3)	6-8
6.1.4 Conditional pdf's for M=4	6-10
6.1.4.1 Single hop/symbol case (L=1)	6-10
6.1.4.2 Two hop/symbol case (L=2)	6-12
6.2 JAMMING EVENTS AND ERROR PROBABILITY FOR L=2	6-16
6.2.1 Jamming Events and Probabilities for M=2	6-16
6.2.2 Jamming Events and Probabilities for M=4	6-19
6.3 AN ALTERNATE APPROACH FOR M=2 AND L=3 HOPS/SYMBOL	6-23
6.4 NUMERICAL RESULTS.	6-25
7.0 COMPARISONS OF RECEIVER PERFORMANCES	7-1
7.1 RECEIVER PERFORMANCES IN FOLLOW-ON NOISE JAMMING	7-1
7.1.1 Formulation of Follow-on Jamming Analysis: Simple Jammer.	7-1
7.1.2 Formulation of FNJ Analysis: Advanced Jammer.	7-2
7.1.3 Performance of Conventional FH/MFSK in FNJ	7-2
7.1.4 Performance of FH/RMFSK in Follow-on Jamming	7-4
7.1.4.1 Soft-decision receivers	7-7
7.1.4.2 Hard-decision receiver	7-13
7.2 COMPARISONS OF FH/RMFSK RECEIVER PERFORMANCES IN WORST-CASE PARTIAL-BAND NOISE JAMMING (WCPBNJ)	7-20

J. S. LEE ASSOCIATES, INC.

TABLE OF CONTENTS (Continued)

	<u>PAGE</u>
7.3 COMPARISONS OF FH/RMFSK AND FH/MFSK	7-35
7.3.1 The M=2, L=1 Case	7-35
7.3.2 L=1 with Alphabet Size Varied	7-38
7.3.3 Cases Where L>1 Hop/Symbol	7-41
7.3.3.1 Linear Combining Receiver	7-42
7.3.3.2 AGC receivers	7-42
7.3.3.3 Clipper receiver	7-42
7.3.3.4 Hard-decision receiver	7-50
7.3.3.5 Self-normalizing receiver	7-51
7.4 COMPARISON OF RECEIVER DIVERSITY EFFECTS	7-56
8.0 ECCM RECEIVER IMPLEMENTATION STUDIES	8-1
8.1 IMPLEMENTATION ISSUES AND CONCEPTS	8-1
8.1.1 ECCM Receiver Information Requirements	8-2
8.1.2 Measurement Approaches	8-5
8.1.2.1 A look-ahead measurement scheme	8-6
8.1.2.2 In-band measurement schemes	8-11
8.1.3 Jamming State Decisions	8-15
8.1.3.1 Jammed channel detection	8-16
8.1.3.2 Jammed symbol detection	8-19
8.2 ASSESSMENT OF IMPLEMENTATION EFFECTS	8-22
8.2.1 Methodology for Direct Assessment	8-22
8.2.2 In Search of an Upper Bound: Simplified Measurement Models	8-25

J. S. LEE ASSOCIATES, INC.

TABLE OF CONTENTS (Continued)

	<u>PAGE</u>
8.2.3 Example Evaluations of Practical AGC Receivers	8-28
8.2.3.1 Analysis of the PACJ receiver	8-28
8.2.3.2 Analysis of the PIC receiver	8-30
8.3 CONCLUSIONS AND RECOMMENDATIONS	8-36
8.3.1 Knowledge Gained from Study	8-36
8.3.1.1 Ideal receiver performances	8-36
8.3.1.2 Practical receiver performances	8-39
8.3.1.3 RMFSK vs MFSK	8-40
8.3.2 Recommendations	8-41
APPENDIX A: PROBABILITY DENSITY FUNCTIONS FOR SOFT DECISION RECEIVER STATISTICS	A-1
APPENDIX B: COMBINATORIAL RELATIONS FOR JAMMING EVENT ENUMERATION	B-1
B.1 NUMBERS OF ORDERED VECTORS	B-1
B.2 NUMBERS OF PARTITIONED ORDERED VECTORS	B-2
B.3 TOTAL NUMBER OF VECTORS REPRESENTED BY ORDERED VECTORS	B-3
B.4 NUMBERS OF ACJ-AGC JAMMING EVENTS	B-5
B.4.1 Number of $\{\ell_0, \underline{\ell}\}$ events	B-5
B.4.2 Summation over ℓ_0 events	B-7
B.4.3 Summation over $\underline{\ell}$ events	B-7
APPENDIX C: DERIVATION OF ERROR RATE EXPRESSIONS FOR "PRACTICAL ACJ" AND "PRACTICAL IC" RECEIVERS	C-1
C.1 JOINT PDF FOR ONE PAIR OF SAMPLES (PACJ)	C-1
C.2 JOINT PDF FOR SUMS OF TWO SAMPLES (PACJ)	C-3

J. S. LEE ASSOCIATES, INC.

TABLE OF CONTENTS (Continued)

	<u>PAGE</u>
C.3 ERROR PROBABILITY FOR L=2 (PACJ)	C-3
C.4 JOINT PDF FOR ONE PAIR OF SAMPLES (PIC)	C-6
C.5 CDF FOR SUM OF TWO NON-SIGNAL VARIABLES (PIC)	C-6
C.6 ERROR PROBABILITY FOR L=2 (PIC)	C-7
APPENDIX D: COMPUTER PROGRAM FOR SQUARE-LAW LINEAR COMBINING RECEIVER	D-1
APPENDIX E: COMPUTER PROGRAM FOR INDIVIDUAL CHANNEL ADAPTIVE GAIN CONTROL RECEIVER	E-1
APPENDIX F: COMPUTER PROGRAM FOR NUMERICAL COMPUTATIONS FOR THE ANY-CHANNEL-JAMMED ADAPTIVE GAIN CONTROL RECEIVER	F-1
APPENDIX G: COMPUTER PROGRAM FOR PLOTTING GRAPHICAL RESULTS FOR THE ANY-CHANNEL-JAMMED ADAPTIVE GAIN CONTROL RECEIVER	G-1
APPENDIX H: COMPUTER PROGRAM FOR CLIPPER RECEIVER WITH M=2 AND L=2	H-1
APPENDIX I: COMPUTER PROGRAM FOR CLIPPER RECEIVER FOR M=4 AND L=2	I-1
APPENDIX J: COMPUTER PROGRAM FOR CLIPPER RECEIVER WITH L=3 HQPS/SYMBOL	J-1
APPENDIX K: COMPUTER PROGRAM FOR SELF-NORMALIZING RECEIVER WITH M=2, L=2	K-1
APPENDIX L: COMPUTER PROGRAM FOR SELF-NORMALIZING RECEIVER WITH M=2, L=3	L-1
APPENDIX M: COMPUTER PROGRAM FOR PRACTICAL ADAPTIVE GAIN CONTROL RECEIVER	M-1
REFERENCES	R-1

J. S. LEE ASSOCIATES, INC.

LIST OF FIGURES

<u>Figures</u>		<u>Pages</u>
1.0-1	COMMUNICATIONS ECCM STRATEGY EVALUATION SCHEME	1-2
1.1-1	THERMAL NOISE AND PARTIAL-BAND NOISE JAMMING MODEL	1-5
1.1-2	TYPICAL L HOPS/SYMBOL FH/MFSK WAVEFORM PATTERN	1-6
1.1-3	TYPICAL L HOPS/SYMBOL FH/RMFSK WAVEFORM PATTERN	1-8
1.1-4	COMPARISON OF THE EXACT ANALYSES WITH THE UNION BOUNDS FOR THE SQUARE-LAW LINEAR COMBINING RECEIVER	1-12
1.1-5	COMPARISON OF THE EXACT ANALYSES WITH THE UNION BOUNDS FOR THE AGC RECEIVER	1-13
1.1-6	COMPARISON OF THE EXACT ANALYSES WITH THE UNION BOUNDS FOR THE CLIPPER RECEIVER	1-14
1.2-1	FH/MFSK SQUARE-LAW GENERIC RECEIVER	1-18
1.2-2	OPTIMUM JAMMING FRACTION (γ) VS. NUMBER OF HOPS/SYMBOL (L) FOR THE AGC AND CLIPPER FH/MFSK (M=8) RECEIVERS WHEN $E_b/N_0 = 9.09$ dB WITH E_b/N_J AS A PARAMETER (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)	1-21
1.2-3	OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE OF FH/MFSK (M=8) SQUARE-LAW COMBINING RECEIVERS FOR L=2 HOPS/SYMBOL WHEN $E_b/N_0 = 9.09$ dB (FOR IDEAL MFSK (M=8) CURVE THE ABSCISSA READS E_b/N_0)	1-22
1.2-4	OPTIMUM JAMMING PERFORMANCE OF THE AGC FH/MFSK (M=4) RECEIVER WHEN $E_b/N_0 = 13.16$ dB WITH THE NUMBER OF HOPS/SYMBOL (L) AS A PARAMETER (FOR IDEAL MFSK (M=4) CURVE THE ABSCISSA READS E_b/N_0)	1-23
1.2-5	OPTIMUM JAMMING PERFORMANCE OF THE AGC RECEIVER FOR BFSK/FH WITH THE NUMBER OF HOPS/BIT (L) AS A PARAMETER WHEN THERMAL NOISE IS ABSENT (FOR IDEAL BFSK CURVE THE ABSCISSA READS E_b/N_0)	1-24

J. S. LEE ASSOCIATES, INC.

LIST OF FIGURES (continued)

<u>Figures</u>	<u>Pages</u>
1.2-6 PROBABILITY OF BIT ERROR VS. E_b/N_0 FOR AGC FH/BFSK RECEIVER USING OPTIMUM NUMBER (L) OF HOPS/BIT, FOR DIFFERENT VALUES OF E_b/N_0 IN WORST-CASE PARTIAL-BAND NOISE JAMMING	1-26
1.3-1 SOFT-DECISION RECEIVER FOR FH/RMFSK	1-28
1.3-2 PARTIAL-BAND JAMMING OF FH/RMFSK	1-30
1.3-3 HARD-DECISION RECEIVER FOR FH/RMFSK	1-33
2.1-1 GENERIC RECEIVER FOR FH/RMFSK	2-2
2.2-1 PROGRAM STRUCTURE FOR CALCULATING JAMMING EVENT PROBABILITIES INDIRECTLY	2-19
3.2-1 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH M=2 AND L=1 HOP/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.3525$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	3-21
3.2-2 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH M=2 AND L=2 HOPS/ SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.3525$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	3-22
3.2-3 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH M=2 AND L=3 HOPS/ SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.3525$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	3-23
3.2-4 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH M=2 AND L=4 HOPS/ SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.3525$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	3-24

J. S. LEE ASSOCIATES, INC.

LIST OF FIGURES (continued)

<u>Figures</u>		<u>Pages</u>
3.2-5	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER AND M=2 WITH NUMBER OF HOPS/BIT AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING	3-25
3.2-6	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH M=4 AND L=1 HOP/ SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	3-27
3.2-7	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH M=4 AND L=2 HOPS/ SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	3-28
3.2-8	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH M=4 AND L=3 HOPS/ SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	3-29
3.2-9	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH M=4 AND L=4 HOPS/ SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	3-30
3.2-10	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER AND M=4 WITH NUMBER OF HOPS/SYMBOL AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING	3-31
3.2-11	WORST-CASE PARTIAL-BAND NOISE JAMMING PERFORMANCE OF FH/RMFSK RECEIVERS FOR L=1 HOP/SYMBOL AND M = 2, 4, 8 WHEN E_b/N_0 GIVES A 10^{-5} BER WITHOUT JAMMING	3-32

J. S. LEE ASSOCIATES, INC.

LIST OF FIGURES (continued)

<u>Figures</u>		<u>Pages</u>
3.2-12	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR M=2 IN THE PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING AT $E_b/N_0 = 13.35247$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1)	3-34
3.2-13	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR M=4 IN THE PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING AT $E_b/N_0 = 10.60657$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1)	3-35
3.2-14	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR M=8 IN THE PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING AT $E_b/N_0 = 9.09401$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1)	3-36
3.2-15	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR M=2 IN THE PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING AT $E_b/N_0 = 20$ dB (FOR MINIMIZATION OF THERMAL NOISE EFFECT)	3-38
3.2-16	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR DQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR M=4 IN THE PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING AT $E_b/N_0 = 20$ dB (FOR MINIMIZATION OF THERMAL NOISE EFFECT)	3-39
3.2-17	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR M=8 IN THE PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING AT $E_b/N_0 = 2-$ dB (FOR MINIMIZATION OF THERMAL NOISE EFFECT)	3-40

J. S. LEE ASSOCIATES, INC.

LIST OF FIGURES (continued)

<u>Figures</u>		<u>Pages</u>
3.2-18	OPTIMUM NUMBER OF HOPPING SLOTS JAMMED (Q_{\max}) THAT PRODUCES $P(e)_{\max}$ VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR THE SQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR M=2 AT $E_b/N_0 = 13.35247$ dB (FOR 10^{-5} BER WHEN L=1)	3-42
3.2-19	OPTIMUM NUMBER OF HOPPING SLOTS JAMMED (Q_{\max}) THAT PRODUCES $P(e)_{\max}$ VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR THE SQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR M=4 AT $E_b/N_0 = 10.60657$ dB (FOR 10^{-5} BER WHEN L=1)	3-43
3.2-20	OPTIMUM NUMBER OF HOPPING SLOTS JAMMED (Q_{\max}) THAT PRODUCES $P(e)_{\max}$ VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR THE SQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR M=8 AT $E_b/N_0 = 9.09401$ dB (FOR 10^{-5} BER WHEN L=1)	3-44
3.2-21	OPTIMUM NUMBER OF HOPPING SLOTS JAMMED (Q_{\max}) THAT PRODUCES $P(e)_{\max}$ VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR THE SQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR M=2 AT $E_b/N_0 = 20$ dB (FOR MINIMIZATION OF THERMAL NOISE)	3-46
4.1-1	POSSIBLE AGC NORMALIZATION SCHEMES	4-3
4.4-1	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH M=2 AND L=1 HOP/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.352471$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-19
4.4-2	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH M=2 AND L=2 HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.352471$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-20
4.4-3	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH M=2 AND L=3 HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.352471$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-21

J. S. LEE ASSOCIATES, INC.

LIST OF FIGURES (continued)

<u>Figures</u>		<u>Pages</u>
4.4-4	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH M=2 AND L=4 HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.352471$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-22
4.4-5	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER AND M=2 WITH NUMBER OF HOPS/SYMBOL L AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING	4-23
4.4-6	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH M=4 AND L=1 HOP/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-24
4.4-7	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH M=4 AND L=2 HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-25
4.4-8	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH M=4 AND L=3 HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-26
4.4-9	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH M=4 AND L=4 HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-27
4.4-10	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER AND M=4 WITH NUMBER OF HOPS/SYMBOL L AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING	4-29

J. S. LEE ASSOCIATES, INC.

LIST OF FIGURES (continued)

<u>Figures</u>	<u>Pages</u>
4.4-11 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH M=8 AND L=1 HOP/ SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 9.094011$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-30
4.4-12 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH M=8 AND L=2 HOPS/ SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 9.094011$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-31
4.4-13 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH M=8 AND L=3 HOPS/ SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 9.094011$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-32
4.4-14 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH M=8 AND L=4 HOPS/ SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 9.094011$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-33
4.4-15 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER AND M=8 WITH NUMBER OF HOPS/SYMBOL AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING	4-34
4.4-16 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER WITH M=2 AND L=1 HOP/ SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.352471$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-36

J. S. LEE ASSOCIATES, INC.

LIST OF FIGURES (continued)

<u>Figures</u>		<u>Pages</u>
4.4-17	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER WITH M=2 AND L=2 HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.352471$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-37
4.4-18	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER WITH M=2 AND L=3 HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.352471$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-38
4.4-19	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER WITH M=2 AND L=4 HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.352471$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-39
4.4-20	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER AND M=2 WITH NUMBER OF HOPS/BIT AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING	4-40
4.4-21	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER WITH M=4 AND L=1 HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-41
4.4-22	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER WITH M=4 AND L=2 HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-42
4.4-23	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER WITH M=4 AND L=3 HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-43

J. S. LEE ASSOCIATES, INC.

LIST OF FIGURES (continued)

<u>Figures</u>		<u>Pages</u>
4.4-24	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER WITH M=4 AND L=4 HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-44
4.4-25	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER AND M=4 WITH NUMBER OF HOPS/SYMBOL AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING	4-45
4.4-26	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER AND M=8 AND L=1 HOP/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 9.094011$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-46
4.4-27	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER AND M=8 AND L=2 HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 9.094011$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER	4-47
4.4-28	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER AND M=8 WITH NUMBER OF HOPS/SYMBOL AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING	4-48
5.0-1	SOFT-DECISION FH/RMFSK RECEIVER WITH CLIPPERS	5-2
5.1-1	EFFECTS OF CLIPPING ON pdf	5-4
5.3-1	PERFORMANCE OF CLIPPER RECEIVER FOR FH/RMFSK WITH M=2, L=2 HOPS/SYMBOL, AND $E_b/N_0 = 13.35247$ dB (FOR $P_b(e) = 10^{-5}$ IN THE ABSENCE OF JAMMING WHEN L=1)	5-32

J. S. LEE ASSOCIATES, INC.

LIST OF FIGURES (continued)

<u>Figures</u>	<u>Pages</u>
5.3-2 PERFORMANCE OF CLIPPER RECEIVER FOR FH/RMFSK WHEN L=2 HOPS/SYMBOL WITH M AS A PARAMETER AND E_b/N_0 CORRESPONDING TO $P_b(e)=10^{-5}$ IN THE ABSENCE OF JAMMING (WHEN L=1 HOP/SYMBOL) . . .	5-33
6.1-1 REGION CONTAINING DECISION VARIABLES	6-13
6.1-2 TWO- AND THREE-DIMENSIONAL CONVOLUTIONS	6-14
6.2-1 VOLUME OF INTEGRATION FOR CORRECT SYMBOL DECISION	6-22
6.4-1 PERFORMANCE OF SELF-NORMALIZING RECEIVER FOR FH/RMFSK WITH M=2, L=2 HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0=13.352471$ dB (FOR $P_b(e)=10^{-5}$ IN THE ABSENCE OF JAMMING WHEN L=1)	6-26
6.4-2 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING NOISE DENSITY RATIO FOR SELF-NORMALIZING RECEIVER IN WORST-CASE PARTIAL-BAND NOISE JAMMING FOR M=2, L=2 AT $E_b/N_0=20$ dB	6-27
6.4-3 PERFORMANCE OF SELF-NORMALIZING RECEIVER FOR FH/RMFSK WITH M=2, L=3 HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0=13.35247$ dB (FOR $P_b(e)=10^{-5}$ IN THE ABSENCE OF JAMMING WHEN L=1)	6-29
6.4-4 PERFORMANCE OF SELF-NORMALIZING RECEIVER FOR FH/RMFSK WITH M=2, AND $E_b/N_0=13.35247$ dB WITH NUMBER OF HOPS/SYMBOL AS A PARAMETER	6-30
7.1-1 THERMAL NOISE AND FOLLOW-ON JAMMER MODELS	7-3
7.1-2 FOLLOW-ON JAMMING PERFORMANCE OF LCR AND ACJ-AGC RECEIVERS FOR FH/RMFSK WITH L=1,2,6 AND M=2,32 WHEN E_b/N_0 YIELDS A 10^{-5} BER FOR L=1	7-5
7.1-3 ADVANCED FOLLOW-ON NOISE JAMMING PERFORMANCE OF FH/RMFSK AND FH/MFSK FOR L=1 HOP/SYMBOL AND UNJAMMED BER = 10^{-5}	7-6
7.1-4 PERFORMANCE OF FH/RMFSK RECEIVERS IN SIMPLE FOLLOW-ON NOISE JAMMING VS. EFFECTIVE BIT ENERGY TO JAMMING NOISE DENSITY RATIO FOR L=1 HOP/SYMBOL AND M=2,4,8 WHEN E_b/N_0 YIELDS A 10^{-5} BER WITHOUT JAMMING	7-10

J. S. LEE ASSOCIATES, INC.

LIST OF FIGURES (continued)

<u>Figures</u>	<u>Pages</u>
7.1-5 PERFORMANCE OF FH/RMFSK RECEIVERS IN SIMPLE FOLLOW-ON NOISE JAMMING VS. EFFECTIVE BIT ENERGY TO JAMMING NOISE DENSITY RATIO FOR L=2 HOPS/SYMBOL AND M=2,4,8 WHEN E_b/N_0 YIELDS A 10^{-5} BER WITHOUT JAMMING	7-12
7.1-6 PERFORMANCE OF HARD-DECISION FH/RMFSK RECEIVER IN SIMPLE FNJ VS. EFFECTIVE E_b/N_j FOR M=2 AND E_b/N_0 SUCH THAT UNJAMMED BER IS 10^{-5} WHEN L=1	7-14
7.1-7 PERFORMANCE OF HARD-DECISION FH/RMFSK RECEIVER IN SIMPLE FNJ VS. EFFECTIVE E_b/N_j FOR M=4 AND E_b/N_0 SUCH THAT UNJAMMED BER IS 10^{-5} WHEN L=1	7-15
7.1-8 PERFORMANCE OF HARD-DECISION FH/RMFSK RECEIVER IN SIMPLE FNJ VS. EFFECTIVE E_b/N_j FOR M=8 AND E_b/N_0 SUCH THAT UNJAMMED BER IS 10^{-5} WHEN L=1	7-16
7.1-9 PERFORMANCE OF HARD-DECISION FH/RMFSK RECEIVER IN SIMPLE FNJ VS. EFFECTIVE E_b/N_j FOR M=2 AND $E_b/N_0=20$ dB	7-17
7.1-10 PERFORMANCE OF HARD-DECISION FH/RMFSK RECEIVER IN SIMPLE FNJ VS. EFFECTIVE E_b/N_j FOR M=4 AND $E_b/N_0=20$ dB	7-18
7.1-11 PERFORMANCE OF HARD-DECISION FH/RMFSK RECEIVER IN SIMPLE FNJ VS. EFFECTIVE E_b/N_j FOR M=8 AND $E_b/N_0=20$ dB	7-19
7.2-1 COMPARISON OF DIFFERENT RECEIVERS FOR FH/RMFSK WITH M=2 AND L=2 HOPS/SYMBOL WHEN $E_b/N_0=13.35247$ dB	7-22
7.2-2 COMPARISON OF DIFFERENT RECEIVERS FOR FH/RMFSK WITH M=2 AND L=2 HOPS/SYMBOL WHEN $E_b/N_0=13.35247$ dB	7-23
7.2-3 COMPARISON OF DIFFERENT RECEIVERS FOR FH/RMFSK WITH M=4 AND L=2 HOPS/SYMBOL WHEN $E_b/N_0=10.60657$ dB	7-24
7.2-4 COMPARISON OF DIFFERENT RECEIVERS FOR FH/RMFSK WITH M=4 AND L=3 HOPS/SYMBOL WHEN $E_b/N_0=10.60657$ dB	7-25
7.2-5 COMPARISON OF DIFFERENT RECEIVERS FOR FH/RMFSK WITH M=2 AND L=2 HOPS/SYMBOL WHEN $E_b/N_0=20$ dB	7-30

J. S. LEE ASSOCIATES, INC.

LIST OF FIGURES (continued)

<u>Figures</u>	<u>Pages</u>
7.2-6 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING NOISE DENSITY RATIO FOR IC-AGC RECEIVER IN WORST-CASE PARTIAL-BAND NOISE FOR M=2 AND L=2 WHEN $E_b/N_0 = 20$ dB	7-32
7.2-7 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING NOISE DENSITY RATIO FOR SELF-NORMALIZING RECEIVER IN WORST-CASE PARTIAL-BAND NOISE JAMMING FOR M=2, L=2 AT $E_b/N_0 = 20$ dB	7-33
7.2-8 WORST-CASE PARTIAL-BAND NOISE JAMMING PERFORMANCE OF THE SELF-NORMALIZING FH/BFSK RECEIVER WHEN THERMAL NOISE IS ABSENT, WITH THE NUMBER OF HOPS/BIT (L) AS A PARAMETER	7-34
7.3-1 COMPARISON OF PERFORMANCE OF ALL RECEIVERS FOR FH/MFSK AND FH/RMFSK WITH M=2 AND L=1 WHEN $E_b/N_0 = 13.35247$ dB	7-36
7.3-2 PERFORMANCE OF INDIVIDUAL CHANNEL AGC RECEIVER FOR FH/RMFSK WHEN L=1 HOP/SYMBOL WITH M AS A PARAMETER AND E_b/N_0 CORRESPONDING TO $P_b(e) = 10^{-5}$ IN THE ABSENCE OF JAMMING	7-39
7.3-3 WORST-CASE PARTIAL-BAND NOISE JAMMING PERFORMANCE OF FH/RMFSK RECEIVERS FOR L=1 HOP/SYMBOL AND M=2,4,8 WHEN E_b/N_0 GIVES A 10^{-5} BER WITHOUT JAMMING	7-40
7.3-4 COMPARISON OF SQUARE-LAW LINEAR COMBINING RECEIVERS FOR FH/MFSK AND FH/RMFSK WITH M=2 AND L=2 HOPS/SYMBOL WHEN $E_b/N_0 = 13.35247$ dB	7-43
7.3-5 COMPARISON OF SQUARE-LAW LINEAR COMBINING RECEIVERS FOR FH/MFSK AND FH/RMFSK WITH M=4 AND L=2 HOPS/SYMBOL WHEN $E_b/N_0 = 10.60657$ dB	7-44
7.3-6 COMPARISON OF AGC RECEIVERS FOR FH/MFSK AND FH/RMFSK WITH M=2 AND L=2 HOPS/SYMBOL WHEN $E_b/N_0 = 13.35247$ dB	7-45
7.3-7 COMPARISON OF AGC RECEIVERS FOR FH/MFSK AND FH/RMFSK WITH M=2 AND L=3 HOPS/SYMBOL WHEN $E_b/N_0 = 13.35247$ dB	7-46
7.3-8 COMPARISON OF AGC RECEIVERS FOR FH/MFSK AND FH/RMFSK WITH M=4 AND L=2 HOPS/SYMBOL WHEN $E_b/N_0 = 10.60657$ dB	7-47

J. S. LEE ASSOCIATES, INC.

LIST OF FIGURES (continued)

<u>Figures</u>		<u>Pages</u>
7.3-9	COMPARISON OF AGC RECEIVERS FOR FH/MFSK AND FH/RMFSK WITH M=4 AND L=3 HOPS/SYMBOL WHEN $E_b/N_0 = 10.60657$ dB	7-48
7.3-10	PERFORMANCE OF CLIPPER RECEIVER FOR FH/RMFSK WHEN L=2 HOPS/SYMBOL WITH M AS A PARAMETER AND E_b/N_0 CORRESPONDING TO $P_b(e) = 10^{-5}$ IN THE ABSENCE OF JAMMING (WHEN L=1 HOP/SYMBOL)	7-49
7.3-11	ERROR PERFORMANCE FOR HARD SYMBOL DECISION FH/RMFSK RECEIVER IN WORST-CASE PARTIAL-BAND NOISE JAMMING WHEN L=3 HOPS/SYMBOL; M=2,4,8; AND E_b/N_0 GIVES 10^{-5} ERROR WITHOUT JAMMING WHEN L=1	7-52
7.3-12	WORST-CASE PARTIAL-BAND NOISE JAMMING PERFORMANCES OF FH/RMFSK AND FH/MFSK USING THE SELF-NORMALIZING RECEIVER, FOR M=2 AND L=2	7-53
7.4-1	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER AND M = 2 WITH NUMBER OF HOPS/BIT AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING	7-57
7.4-2	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER AND M=2 WITH NUMBER OF HOPS/SYMBOL AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING	7-59
7.4-3	OPTIMUM DIVERSITY PERFORMANCE OF INDIVIDUAL CHANNEL AGC RECEIVER FOR FH/RMFSK WITH M=2 AND E_b/N_0 AS A PARAMETER	7-60
7.4-4	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER AND M=2 WITH NUMBER OF HOPS/BIT AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING	7-62
7.4-5	BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING NOISE DENSITY RATIO FOR SELF-NORMALIZING RECEIVER IN WORST-CASE PARTIAL-BAND NOISE FOR M=2 AND $E_b/N_0=13.35$ dB, WITH THE NUMBER OF HOPS/BIT (L) VARIED	7-63

J. S. LEE ASSOCIATES, INC.

LIST OF FIGURES (continued)

<u>Figures</u>		<u>Pages</u>
7.4-6	OPTIMUM DIVERSITY PERFORMANCE OF HARD DECISION RECEIVER FOR FH/RMFSK WITH M=2	7-64
8.1-1	ACQUISITION OF JAMMING STATE DATA BY A "LOOK-AHEAD" SCHEME . . .	8-7
8.1-2	USE OF LOOK-AHEAD MEASUREMENTS TO DERIVE ECCM WEIGHTS	8-8
8.1-3	POSSIBLE USE OF MULTIPLE-HOP/MULTIPLE CHANNEL SAMPLES TO IMPROVE MEASUREMENTS	8-10
8.1-4	CONCEPTUAL PROCESSING OF IN-BAND MEASUREMENTS TO IMPLEMENT AGC RECEIVER	8-13
8.1-5	PROCESSING OF IN-BAND MEASUREMENTS TO IMPLEMENT CLIPPER RECEIVER	8-14
8.2-1	PRACTICAL ACJ-AGC (PACJ) RECEIVER FOR FH/RMFSK WHEN M=2	8-29
8.2-2	BER PERFORMANCE FOR PACJ FH/RMFSK RECEIVER IN PARTIAL-BAND NOISE JAMMING FOR M=2, L=2, AND $E_b/N_0 = 13.35$ dB	8-31
8.2-3	BER PERFORMANCE FOR PACJ FH/RMFSK RECEIVER IN PARTIAL-BAND NOISE JAMMING FOR M=2, L=2, AND $E_b/N_0 = 20$ dB	8-32
8.2-4	PRACTICAL IC-AGC (PIC) RECEIVER FOR FH/RMFSK WHEN M=2	8-33
8.2-5	PERFORMANCE OF THE PRACTICAL IC-AGC FH/RMFSK RECEIVER IN FULL-BAND NOISE JAMMING FOR M=L=2	8-35
8.2-6	PERFORMANCE OF THE PRACTICAL IC-AGC FH/RMFSK RECEIVER USING A TWO-SAMPLE NOISE ESTIMATE IN FULL-BAND NOISE JAMMING FOR M=L=2	8-37

J. S. LEE ASSOCIATES, INC.

LIST OF TABLES

<u>TABLES</u>	<u>PAGE</u>
1.2-1 DESCRIPTIONS OF THE RECEIVERS	1-19
1.3-1 RECEIVER PROCESSING FUNCTIONS STUDIED	1-32
2.2-1 NUMBER OF JAMMING EVENTS	2-8
2.2-2 MINIMUM NUMBER OF DISTINGUISHABLE EVENTS	2-9
2.2-3 JAMMING EVENT PROBABILITIES FOR M=2, L=1 TO 4	2-17
2.2-4 PROBABILITIES OF $\underline{\ell}$ JAMMING EVENTS FOR M = 4	2-18
3.1-1 POLYNOMIALS FOR $F_L(\alpha; \ell)$	3-10
3.1-2 COEFFICIENTS FOR EQUATION (3.1-29)	3-12
3.1-3 EXPRESSION FOR CONDITIONAL ERROR PROBABILITY	3-13
4.3-1 PROBABILITIES OF $(\ell_0, \underline{\ell})$ JAMMING EVENTS FOR M = 4	4-17
5.1-1 NON-SIGNAL CHANNEL PROBABILITY DENSITY FUNCTIONS, L = 1 TO 3	5-12
5.1-2 NON-SIGNAL CHANNEL CUMULATIVE DISTRIBUTION FUNCTIONS, L=1 TO 3	5-14
5.1-3 SIGNAL CHANNEL PROBABILITY DENSITY FUNCTIONS, L = 1,2	5-16
5.1-4 SIGNAL CHANNEL PROBABILITY DENSITY FUNCTIONS, L = 3	5-17
5.3-1 OPTIMUM NORMALIZED CLIPPING THRESHOLD $n_0/2\sigma_N^2$	5-29
6.1-1 L=1 PROBABILITY DENSITIES FOR M=4	6-11
6.2-1 JAMMING EVENTS AND PROBABILITIES FOR M=2, L=2	6-17
6.2-2 JAMMING EVENTS AND PROBABILITIES FOR M=4, L=2	6-20
8.1-1 SUMMARY OF ECCM TECHNIQUES FOR THE RECEIVERS STUDIED	8-3

J. S. LEE ASSOCIATES, INC.

THE EFFECTIVENESS OF RANDOM MFSK FREQUENCY HOPPING ECCM RADIOS AGAINST WORST-CASE PARTIAL-BAND NOISE JAMMING

1.0 INTRODUCTION

The purpose of this study is to provide the Army a direct comparison of the uncoded bit error rate (BER) performance of several receiver anti-jam processing schemes with varying degrees of implementation complexity, under the same conditions of system noise and jamming. In this manner the engineering cost of complex anti-jam receiver designs can be weighed against their effectiveness, as illustrated in Figure 1.0-1. In what follows we discuss the issues surrounding the work and summarize our effort.

1.1 BACKGROUND

In the Electronic Warfare (EW) environment, where a "battle" is waged between the communicating party and the party that is engaged in the pursuit of disrupting the communicator's link, strategy plays an important and fundamental role for the opposing parties. To the communicating party, the opponent's Electronic Support Measures (ESM) and Electronic Counter-measures (ECM) pose as threats. ESM involves essentially activities for spectrum surveillance and direction finding by passive means, whereas ECM involves activities for the purpose of victimizing the communicator's link. Jamming is an active measure of accomplishing ECM objectives. It is, therefore,

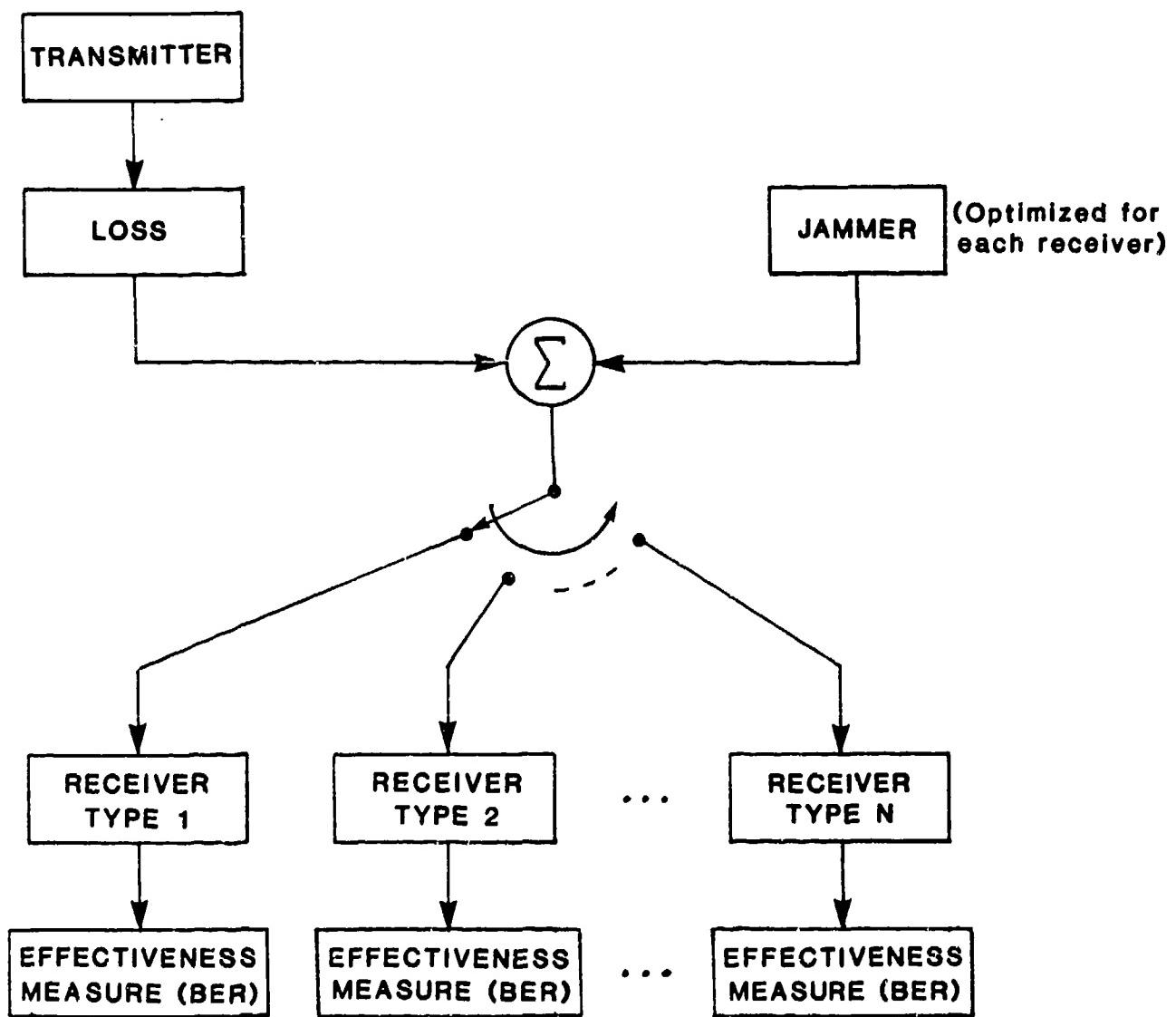


FIGURE 1.0-1 COMMUNICATIONS ECCM STRATEGY EVALUATION SCHEME

J. S. LEE ASSOCIATES, INC.

easy to recognize that fixed-frequency radios are very much vulnerable to ESM and ECM attacks. Communication systems that are designed to counter or mitigate the effects of ESM or ECM attacks are termed Electronic Counter-countermeasures (ECCM) radios, or jam-resistant communication systems.

In principle, there exist many different schemes which can provide the communicator with jam-resistant radio capabilities; Direct Sequence (DS) spread-spectrum and Frequency-Hopping (FH) spread-spectrum systems are two generic schemes. While the DS/SS system requires phase coherence over the system's wide operational bandwidth in its implementation, the FH/SS system does not. The fact that most of the tactical ECCM radios are of FH/SS type is based not only on this reason, but also on the fact that the attainable "processing gain" is achieved with less complexity and cost.

1.1.1 Jamming Strategy Against Frequency-Hopping Radios

ECCM radio designs are based on the desire to suppress the total jamming power by an amount equal to the processing gain, defined as the ratio of FH system bandwidth to the receiver noise bandwidth. The difference between the processing gain in dB and the SNR in dB required for traffic demodulation is the (anti-jam) margin that the communicator can use to tolerate an excess of jammer power over signal power at the system front end. The intelligent jammer, however, does not spread his power over the entire system bandwidth, so that the definition and effects of processing gain will not apply.

The jammer may employ a partial-band noise jamming strategy, in which the available jammer power is placed in a fraction (γ) of the radio

J. S. LEE ASSOCIATES, INC.

system bandwidth, as illustrated in Figure 1.1-1. Assuming that total power is fixed, there is an optimum value of γ which achieves the most effective tradeoff of the probability of jamming and the probability of error when jammed, thus achieving a maximum overall error rate for the given amount of jammer power.

The jammer may, if it is feasible, concentrate his power further if he can intercept the hopping signal in real time and immediately broadcast a strong burst of noise in the frequencies near the signal (follow-on jamming). This type of jamming can be successful against FH systems in which a conventional narrowband communications signal is slowly hopped by simple translation in frequency.

1.1.2 ECCM Waveforms

Along with using hopped signals, the communicator can exercise an additional degree of freedom by employing a low energy density waveform to minimize interceptions by a potential jammer. Such a waveform is FH/MFSK using a number of hops per symbol (L), a kind of repetition code or diversity [5] to permit transmission at lower power and/or to combat fading. The conventional form of FH/MFSK is illustrated in Figure 1.1-2; once a symbol has been chosen for a given interval T_s , a conventional MFSK signal is generated and randomly hopped (translated) L times at a rate $R_H = L/T_s = 1/\tau$ before a new symbol is keyed. Because the M possible symbol frequencies are adjacent, this waveform is vulnerable to follow-on repeat jamming, unless the hopping rate can be made very high.

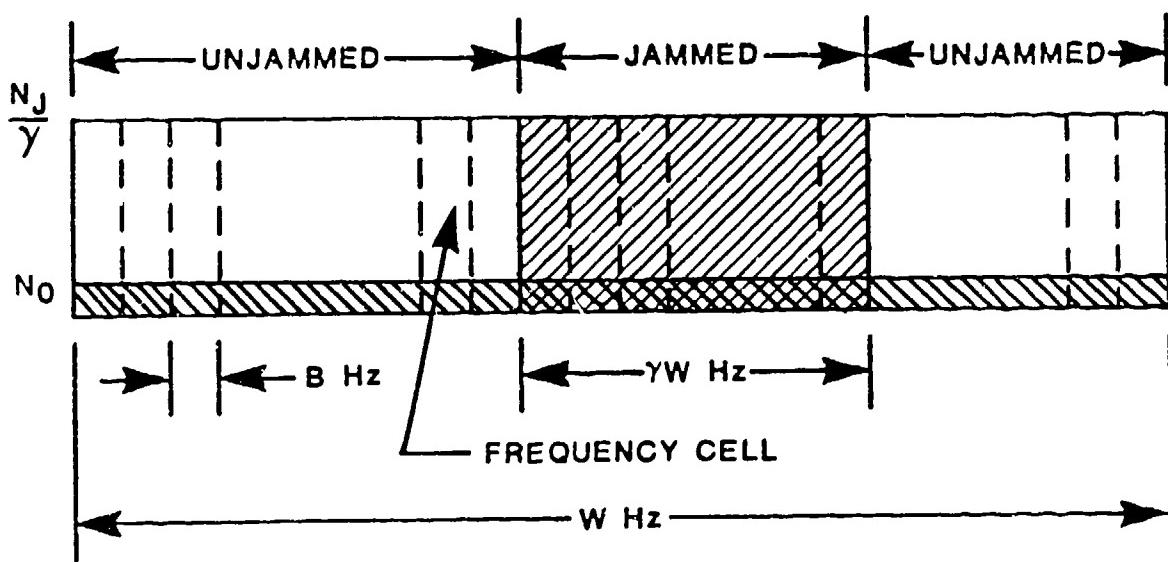


FIGURE 1.1-1 THERMAL NOISE AND PARTIAL-BAND NOISE JAMMING MODEL

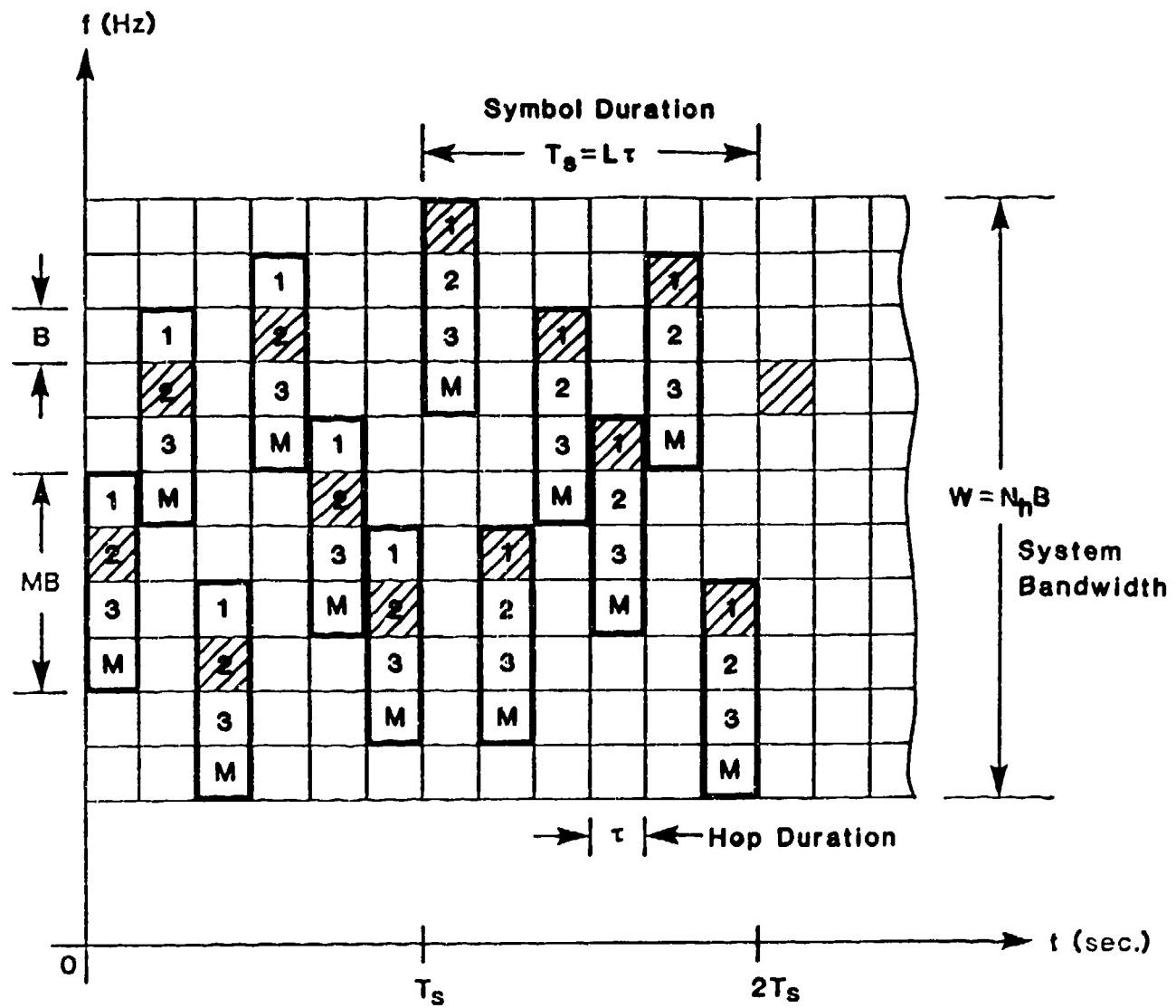


FIGURE 1.1-2 TYPICAL L HOPS/SYMBOL FH/MFSK WAVEFORM PATTERN

If the individual symbol frequencies of the MFSK symbol are assigned randomly on a per hop basis, as illustrated in Figure 1.1-3, then repeat jamming is less likely to produce an error since the symbol frequencies (at RF) are no longer adjacent [6,7]. (We shall refer to this waveform as FH/RMFSK with L hops per symbol.) It has been shown [6] that, for the special case of one hop/bit binary systems ($M=2$) and very little system or thermal noise ($E_b/N_0 = 30$ dB), the two forms of FH/BFSK achieve the same performance in optimum partial-band noise jamming. This suggests that the random MFSK waveform will perhaps be a better choice for $L > 1$ and $M > 2$, although it has not been established as a fact that its performance in partial-band noise is always equal to that of the conventional system, while offering additional protection against follow-on jamming.

The FH/RMFSK implementation is more complex, and the study in this report will permit the cost of this additional complexity to be weighed against its performance, compared to that of the more conventional FH/MFSK as calculated by LAI [1].

1.1.3 Motivation for the Proposed Random Hopping

As we have stated above, the proposed FH/RMFSK waveform is less vulnerable to follow-on or repeat jamming than is a conventional FH/MFSK systems with M contiguous signalling frequencies. In addition, the FH/RMFSK waveform is less vulnerable to tone jamming, since the randomized selection of the M frequencies reduces the amount of structure in the signal. This makes it

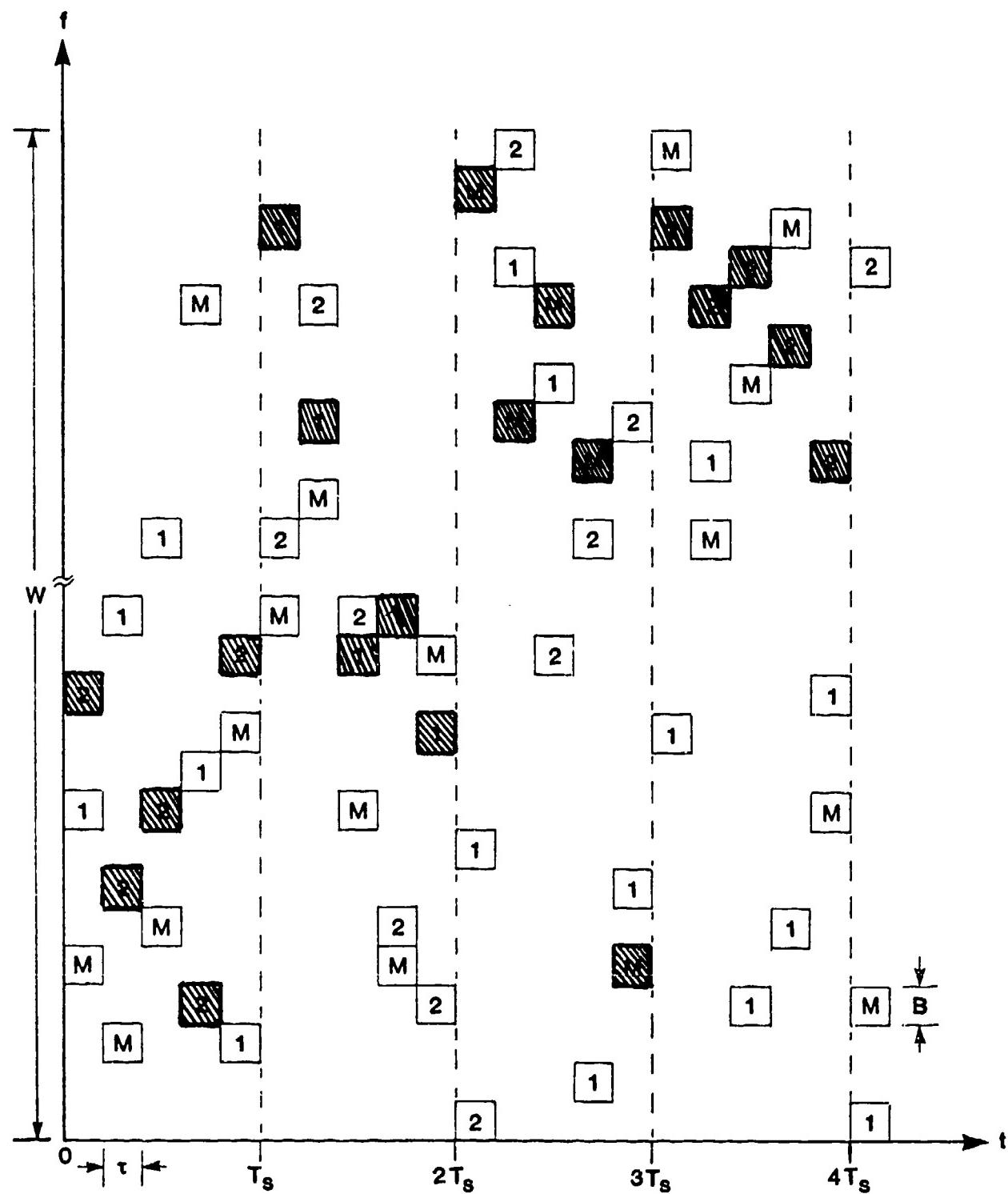


FIGURE 1.1-3 TYPICAL L HOPS/SYMBOL FH/RMFSK WAVEFORM PATTERN

J. S. LEE ASSOCIATES, INC.

much more difficult for a tone jammer to implement an optimum jamming strategy consisting of one jamming tone per M-ary symbol: the lack of structure in the hopping gives the jammer no features to exploit to insure a tone hits the symbol; thus the jammer is forced to divide his available power up into more tones, resulting in a lessened effect on the communications link when it hops into a jammed slot.

Therefore, the motivation for considering the use of FH/RMFSK as an LPI anti-jam communications system design is based upon a desire to lessen or reduce vulnerability to certain more sophisticated jamming threats, such as follow-on jamming and tone jamming. However, before the system can be considered a viable design candidate, its performance under the less sophisticated jamming, namely partial-band noise jamming, must be known.

1.1.4 Rationale for the Exact Analysis of FH/MFSK System Performance In Partial-Band Noise Jamming

It is known that the advantage of an M-ary orthogonal modulation system rests on the fact that the scheme requires less energy per data bit transmission than other available modulation schemes. Cost-effective implementation (efficient non-coherent detection) is another reason in selecting M-ary FSK waveforms by designers of ECCM radios. Recently, Hughes Aircraft Company has conducted studies for U.S. Army CECOM on feasibility of AJ/LPI ECCM techniques, employing L-hops per symbol FH/MFSK [8].

Exact knowledge of performance measures and vulnerability of L-hops per symbol FH/MFSK SS systems has not been available until recently,

J. S. LEE ASSOCIATES, INC.

and that of FH/RMFSK is yet to be determined. Workers in this field previously held the view that M-ary system performance measures could be estimated once the performance measures of the binary systems are available. This view was based on the conventional wisdom of applying the "union bound". As is well known, once we know the binary system performance, that of the M-ary system can be approximated by the union bound given by

$$P_M(e;E_s) \leq (M-1) P_2(e;E_s) \quad (1.1-1)$$

where $P_2(e;E_s)$ is the probability of error for the binary system, using any pair of symbols from the set $\{s_1(t), s_2(t), \dots, s_M(t)\}$, and $P_M(e;E_s)$ is the M-ary system probability of error, where E_s is the symbol energy. The workers have also invoked the well-known relationship between the bit error probability and the symbol (K-bit word) error probability for the M-ary orthogonal system. That is,

$$P_b(e;E_b) = \frac{M}{2(M-1)} P_M(e;E_s). \quad (1.1-2)$$

where $P_b(e;E_b)$ denotes bit error probability and E_b is the energy per bit.

By putting equation (1.1-1) into equation (1.1-2), we obtain the "union bound"-based approximate-performance measure of the bit error probability of an MFSK systems, given by

$$P_b(e;E_b) \leq \frac{M}{2} P_2(e;E_s) = 2^{K-1} P_2(e;KE_b) \quad (1.1-3)$$

where

$$K = \log_2 M. \quad (1.1-4)$$

J. S. LEE ASSOCIATES, INC.

In principle, one can use equation (1.1-3) to assess the performance of an M-ary orthogonal system. Our discovery, however, did not support this generality when the communication channel is of the non-exponential type. In an attempt to obtain an approximate performance measure of 2-hops per symbol FH/MFSK system under partial-band noise jamming environment for $M=4, 8, 16$, and 32 , we have used equation (1.1-3) in applying the binary results, as shown in Figures 1.1-4 to 1.1-6, to three different receiver schemes. A surprising result is that as M is increased, bit error probability as given by the union bound is worsened for all three receivers! Interpretation of this result is that one needs to expend more energy per bit in the higher-order-message-alphabet orthogonal system, a result that is not supportable even on the basis of intuition; and is, indeed, contrary to the exact results shown in the figures.

The above paragraph is to point out that the "union bound" cannot be used when one considers non-Gaussian channels such as partial-band noise, as experienced by a FH/MFSK system. These channels are inverse linear channels, and they do not allow the union bounding techniques to be applicable in assessing M-ary system performances. Thus, one can conclude that exact analysis is necessary.

1.1.5 Extension of Uncoded Error Analysis to Coded Performance

While error-control coding is quite likely to be used by the communicator to counter any jamming effects, the analysis of total system performance may be usefully divided into two parts: uncoded performance and

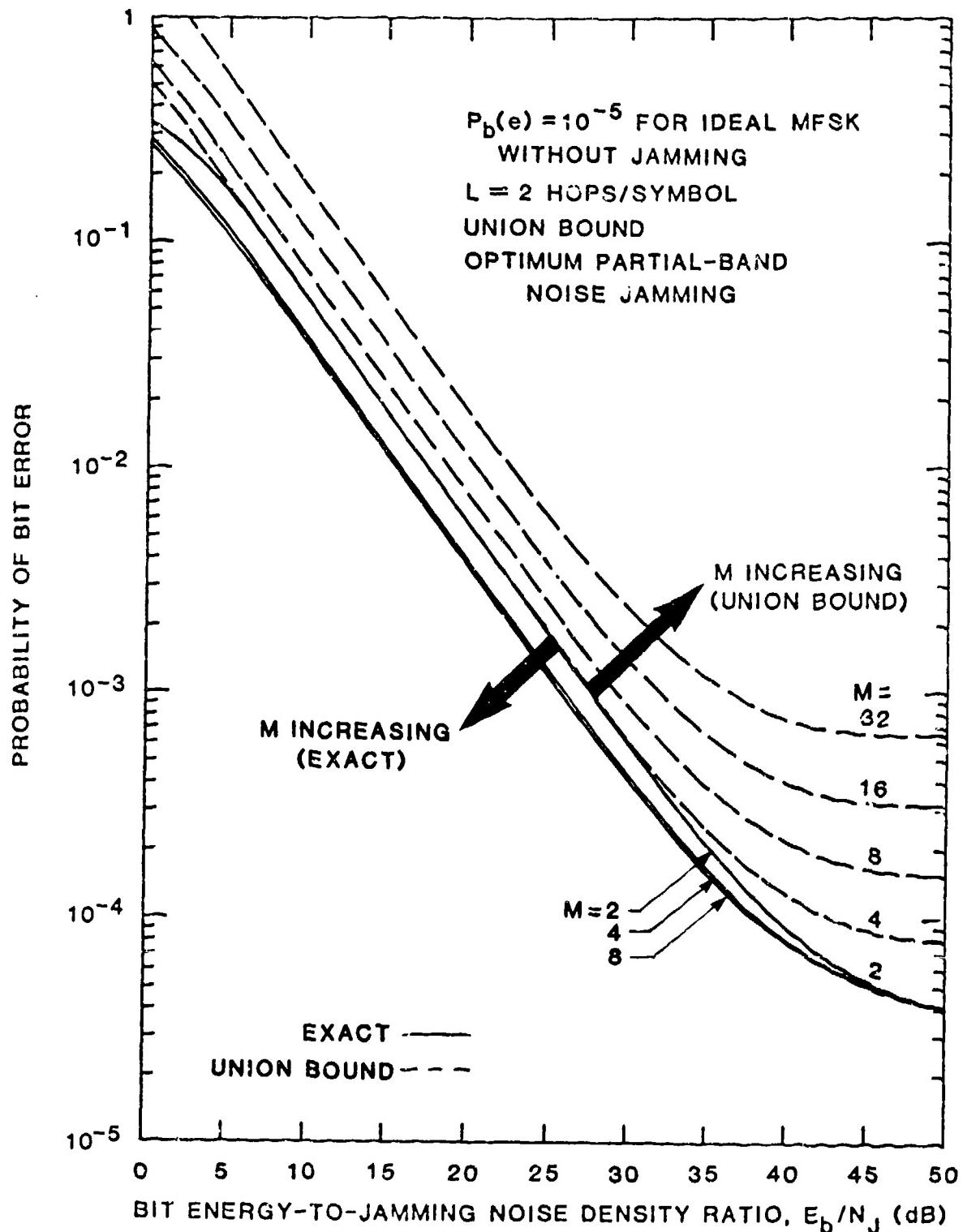


FIGURE 1.1-4 COMPARISON OF THE EXACT ANALYSES WITH THE UNION BOUNDS FOR THE SQUARE-LAW LINEAR COMBINING RECEIVER

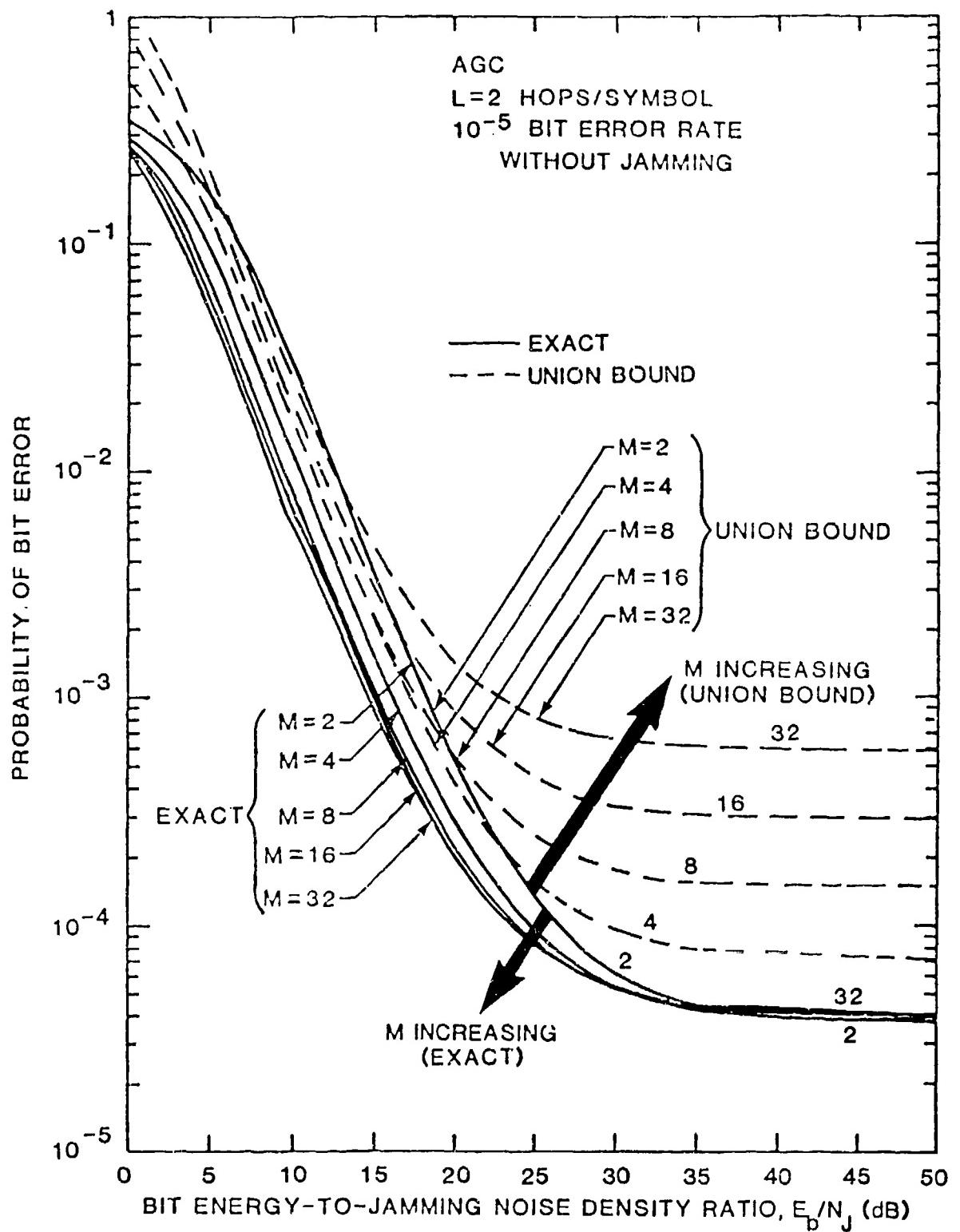


FIGURE 1.1-5 COMPARISON OF THE EXACT ANALYSES WITH THE UNION BOUNDS FOR THE AGC RECEIVER

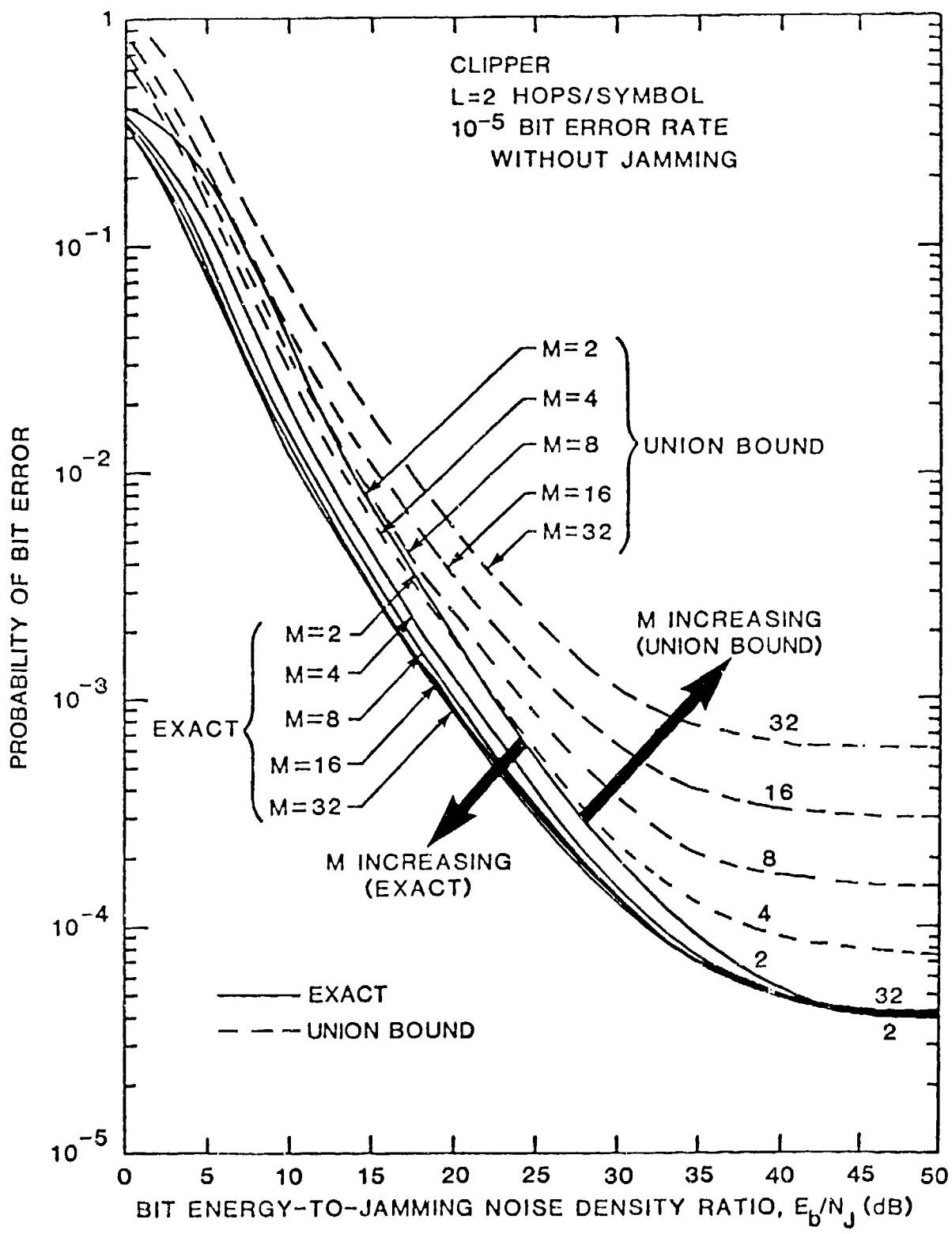


FIGURE 1.1-6 COMPARISON OF EXACT ANALYSES WITH THE UNION BOUNDS FOR THE CLIPPER RECEIVER

enhanced performance using coding. For comparison studies of anti-jam demodulation schemes such as we are proposing, it is sufficient to consider uncoded performance, since the coded system performance is proportional to the uncoded.

For example, for code words using n channel symbols the probability of word error for a bounded-distance decoding algorithm is [9]

$$P_W = \sum_{i=t+1}^n \binom{n}{i} p_s^i (1 - p_s)^{n-i} \quad (1.1-5)$$

where p_s is the uncoded performance in terms of symbol errors and t is the number of correctable errors. This word error probability can be translated into an equivalent information bit error probability by a formula appropriate to the particular coding and decoding algorithms.

1.2 ECCM PROCESSING

Once the FH/MFSK or FH/RMFSK waveform has been dehopped at the intended receiver, the L hops constituting the MFSK symbol can be combined in several ways. It has been shown [10] that the conventional method of summing up the (non-coherent) L hop energies, although effective against fading, produces a BER which increases with L against optimum partial-band noise jamming. Therefore, a number of non-linear combining schemes have been studied, based on weighting the dehopped and envelope-detected hops in some fashion to discriminate against those hops which have been jammed [1, 11, 12].

J. S. LEE ASSOCIATES, INC.

Using these nonlinear combining schemes it has been shown for FH/MFSK that the use of $L > 1$ hops per symbol can be understood as providing a kind of diversity improvement against the jamming, depending on the system noise level.

It has not been determined how a FH/RMFSK waveform with L hops per symbol will perform against optimum partial-band noise, whether using conventional or nonlinear soft-decision combining of the hops.*

1.2.1 Examples of Receiver Effectiveness Computations for FH/MFSK

Under contract to the Office of Naval Research, LAI has studied in great detail the uncoded performances of frequency hopped BFSK and MFSK communication systems under optimum partial-band noise jamming [1, 10, 11, 13, 14, 15]. The focus of these efforts has been to determine both the optimum partial-band jamming strategy and the most effective anti-jam receiver processing schemes for this type of modulation, using exact analyses which include the system's thermal noise. One of the chief results of our work has been the discovery that conclusions drawn from previous, approximate studies neglecting thermal noise are not strictly valid. It had been commonly asserted that the use of multiple hops per symbol in FH/MFSK systems provides a diversity gain improvement against optimum partial-band jamming in much the same way that it does against the effect of fading on the signal. We have been able to show that this improvement does not exist for the conventional (linear combining) receiver, and we have demonstrated quantitatively

*In a recent paper [16], FH/RMFSK performance with L hops per symbol has been shown for a receiver using hard decisions. The binary hard-decision case was also analyzed in [17].

J. S. LEE ASSOCIATES, INC.

that a limited improvement holds for certain nonlinear hop combining receiver processing schemes ("metrics"), as a function of the system's thermal noise.

A generic model of FH/MFSK square-law receivers is given in Figure 1.2-1. Among the processing schemes, represented by the function $f_k(\cdot)$ in the figure prior to the accumulation of soft decision statistics $\{z_m\}$, are those listed in Table 1.2-1. The performance of the conventional, linear combining receiver in optimum partial-band noise jamming was calculated directly and compared to that of the three nonlinear combining receivers. For the calculation, the bit error probabilities were expressed by

$$P_b(e) = \sum_{\ell=0}^L p_\ell P_b(e|\ell), \quad (1.2-1)$$

where p_ℓ is the probability that ℓ out of L hops constituting a given symbol are jammed, and $P_b(e|\ell)$ is the bit error probability given that ℓ hops are jammed. For conventional FH/MFSK, we have assumed that

$$p_\ell = \binom{L}{\ell} \gamma^\ell (1-\gamma)^{L-\ell} \quad (1.2-2)$$

based on all of the M symbol frequency slots being jammed on a given hop, with probability γ (the fraction of the system bandwidth which is jammed), or none of them being jammed, with the probability $1-\gamma$.

For each receiver type and values of E_b/N_0 and E_b/N_J , the maximum bit error probability was found as a function of γ , the partial-band jamming

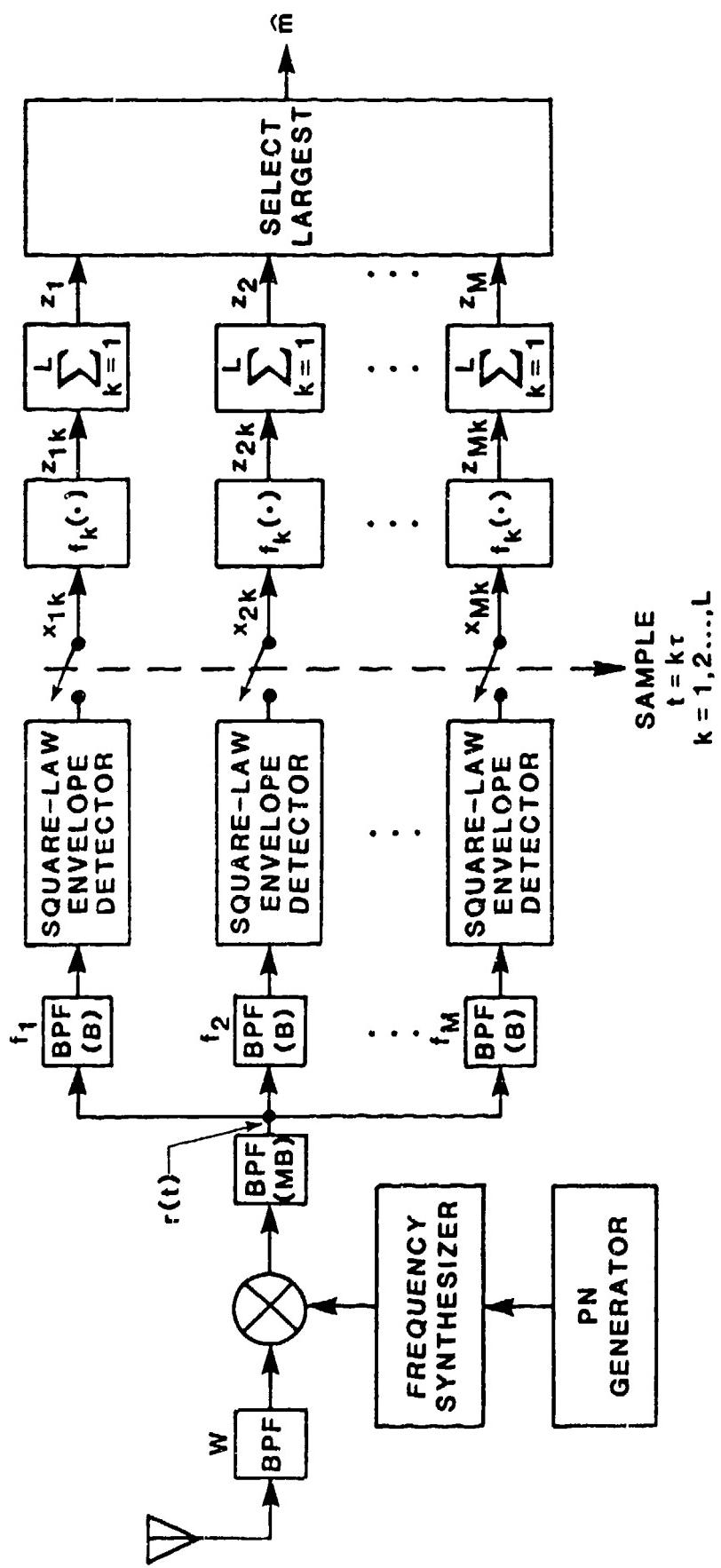


FIGURE 1.2-1 FH/MFSK SQUARE-LAW GENERIC RECEIVER

J. S. LEE ASSOCIATES, INC.

TABLE 1.2-1

DESCRIPTIONS OF THE RECEIVERS

RECEIVER TYPE	SPECIFICATION OF $z_{ik} = f_k(x_{ik}), i=1, 2, \dots, M$	REMARKS
LINEAR COMBINING RECEIVER	$z_{ik} = x_{ik}$	Direct Connection (Linear Combining)
CLIPPER RECEIVER	$z_{ik} = \begin{cases} x_{ik}, & x_{ik} \leq n \\ n, & x_{ik} > n \end{cases}$	Soft Limiter (Nonlinear Combining)
AGC RECEIVER	$z_{ik} = x_{ik}/\sigma_k^2$ $\sigma_k^2 = \begin{cases} \sigma_N^2, & \text{if not jammed} \\ \sigma_N^2 + \sigma_J^2, & \text{if jammed} \end{cases}$ $(\sigma_k^2 = \text{measured})$	Adaptive Gain Control (Nonlinear Combining)
SELF-NORMALIZING RECEIVER	$z_{ik} = \frac{x_{ik}}{\sum_{i=1}^M x_{ik}}$	Practical Realization of AGC Using In-Band Measurements

fraction. These calculations revealed significant differences among the receiver types in the optimum value of γ as well as in the bit error probability. For example, in Figure 1.2-2, we show that for $M=8$ that the jammer's optimum γ is much more sensitive to the value of L , the number of hops per MFSK symbol, for the AGC receiver than for the clipper receiver. Therefore the jammer must have more accurate information on the modulation parameters in order to be as effective as possible against the AGC receiver.

Another typical result is the comparison shown in Figure 1.2-3, also for $M=8$, and for $L=2$ hops per symbol. We see that the (ideal) AGC form of ECCM receiver processing is significantly better at combatting the effects of the jamming, and that the clipper receiver also improves the BER, but not as much.

Figure 1.2-4 shows the effect of increasing E_b/N_0 so as to provide a lower bit error probability in the absence of jamming for the AGC receiver with $M=4$ and L as a parameter. We see that under these conditions, the optimum choice of L includes higher values of the number of hops per symbol before increased noncoherent combining loss dominates and forces a choice of a lower value of L .

Figure 1.2-5 illustrates the performance as $E_b/N_0 \rightarrow \infty$, i.e. no thermal noise, for FH/BFSK (i.e. $M=2$). We see that in the absence of thermal noise, the optimum value of L increases without limit as E_b/N_J increases. A similar result holds for the case of $M>2$.

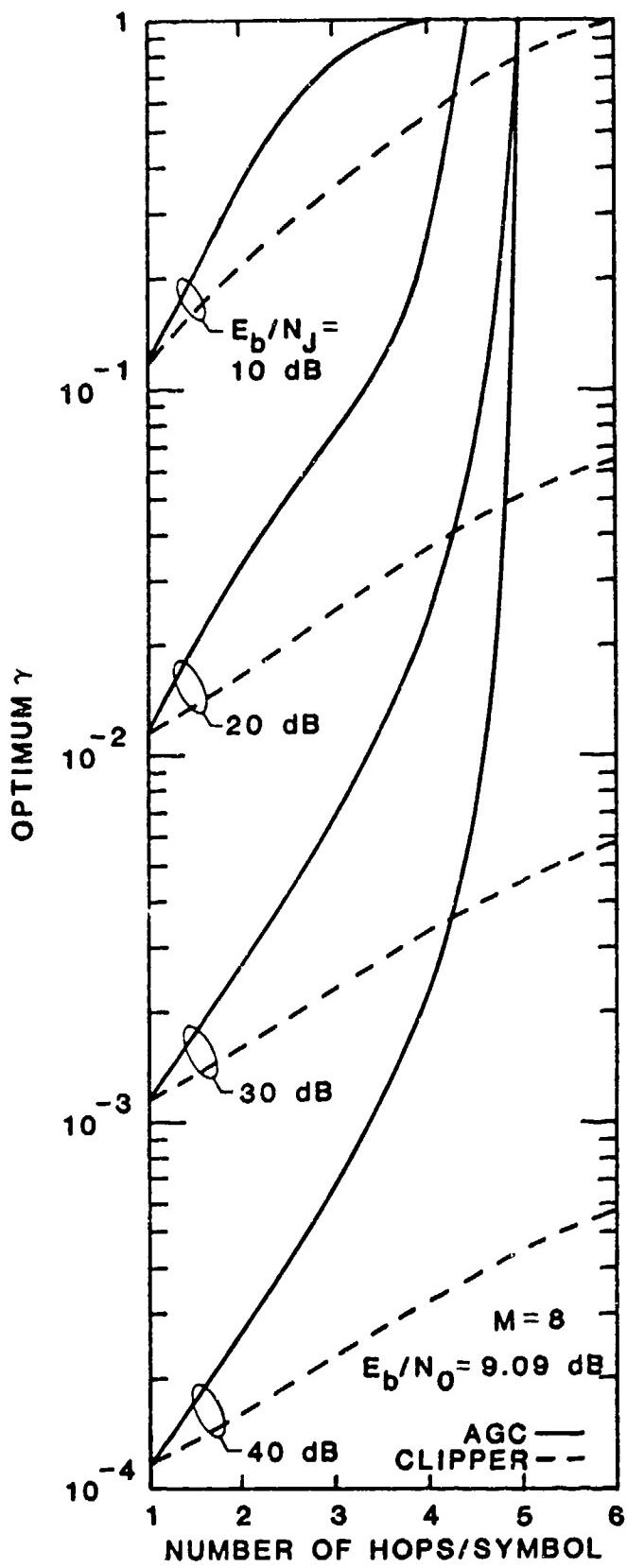


FIGURE 1.2-2 OPTIMUM JAMMING FRACTION (γ) VS. NUMBER OF HOPS/SYMBOL (L) FOR THE AGC AND CLIPPER FH/MFSK ($M = 8$) RECEIVERS WHEN $E_b/N_0 = 9.09$ dB WITH E_b/N_j AS A PARAMETER (FOR 10^{-5} ERROR RATE WITHOUT JAMMING)

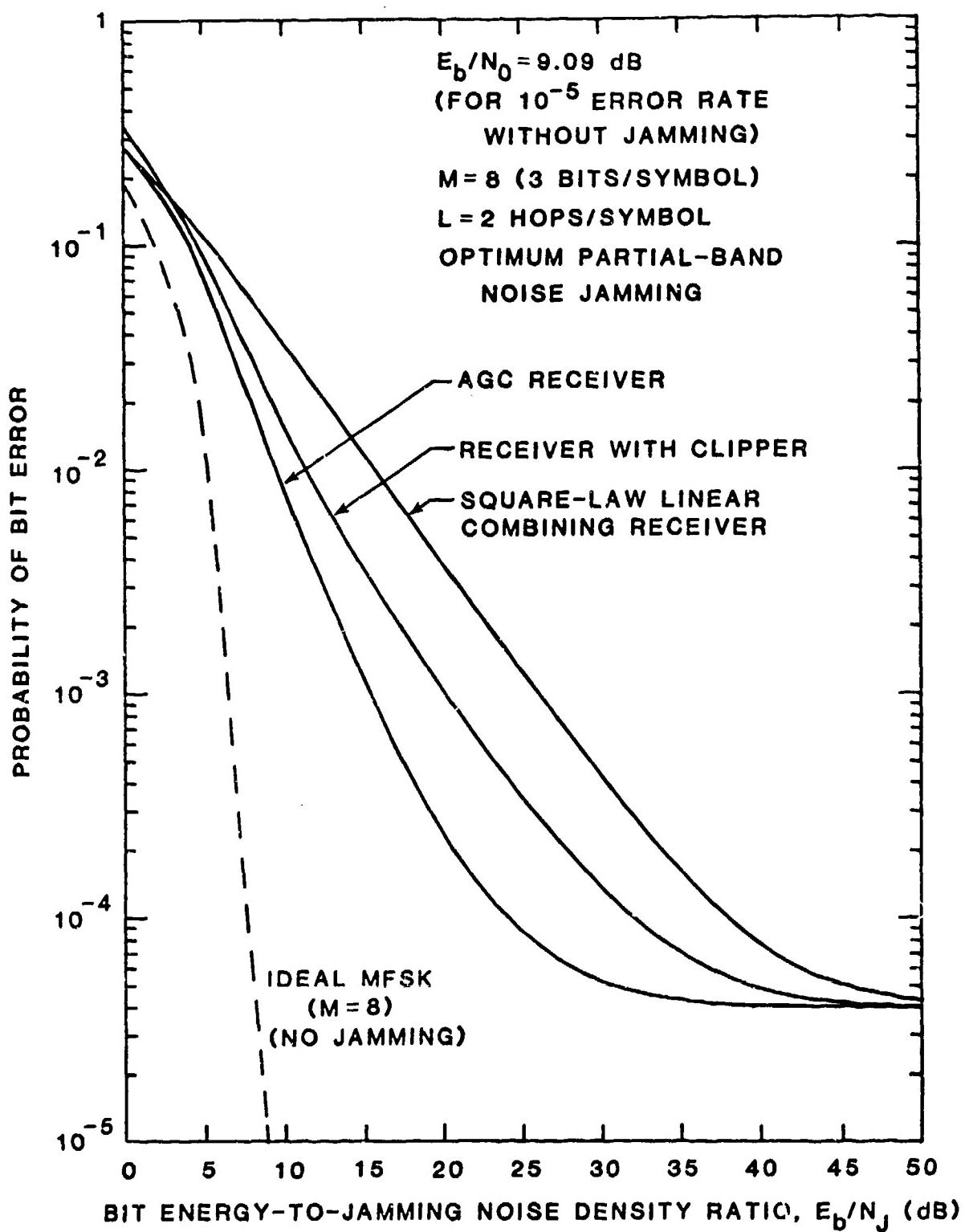


FIGURE 1.2-3 OPTIMUM PARTIAL-BAND NOISE JAMMING PERFORMANCE
 OF FH/MFSK ($M = 8$) SQUARE-LAW COMBINING RECEIVERS
 FOR $L = 2$ HOPS/SYMBOL WHEN $E_b/N_0 = 9.09$ dB (FOR IDEAL
 MFSK ($M = 8$) CURVE THE ABSCISSA READS E_b/N_0)

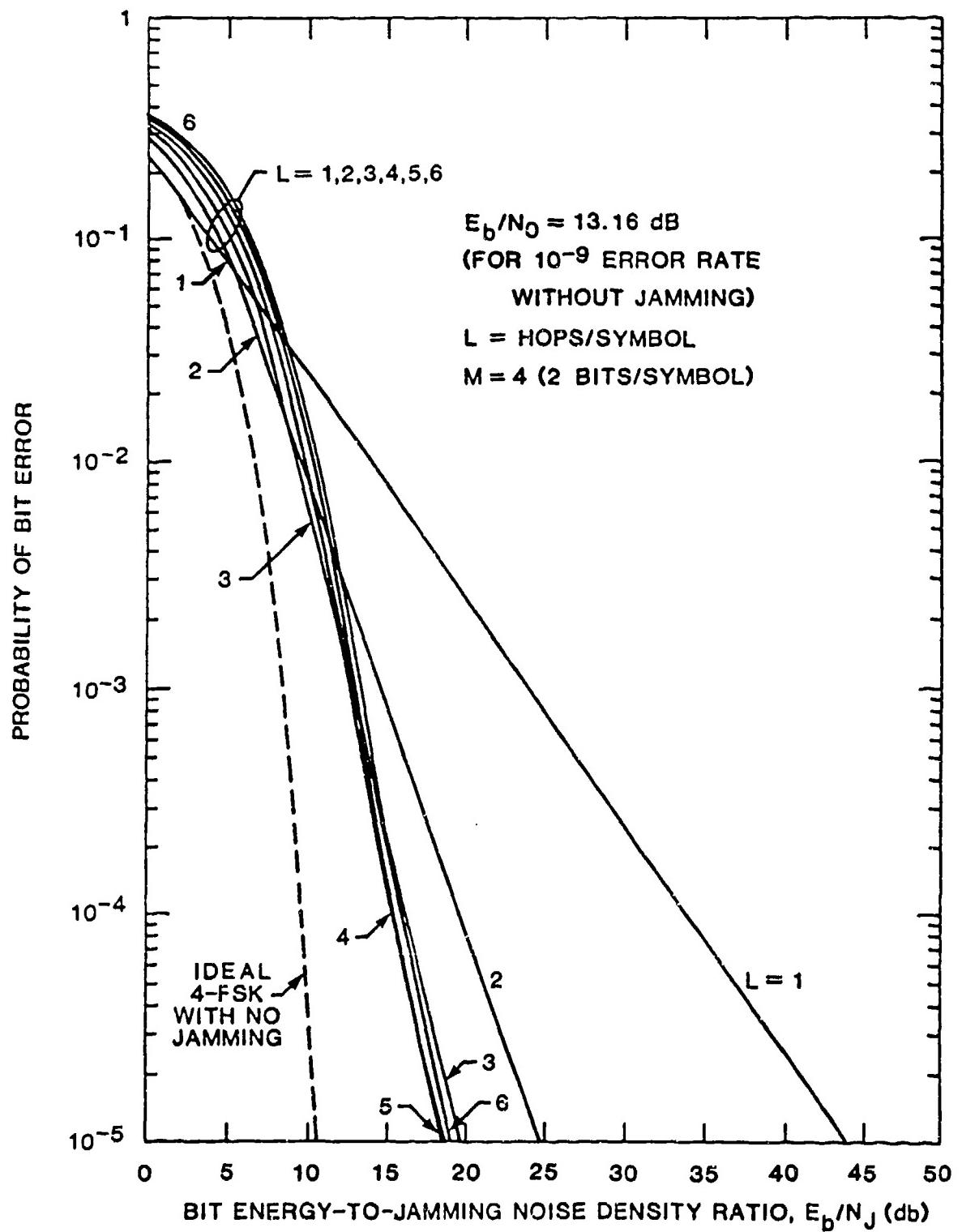


FIGURE 1.2-4 OPTIMUM JAMMING PERFORMANCE OF THE AGC FH/MFSK ($M = 4$) RECEIVER WHEN $E_b/N_0 = 13.16 \text{ dB}$ WITH THE NUMBER OF HOPS/SYMBOL (L) AS A PARAMETER (FOR IDEAL MFSK ($M = 4$) CURVE THE ABSCISSA READS E_b/N_0)

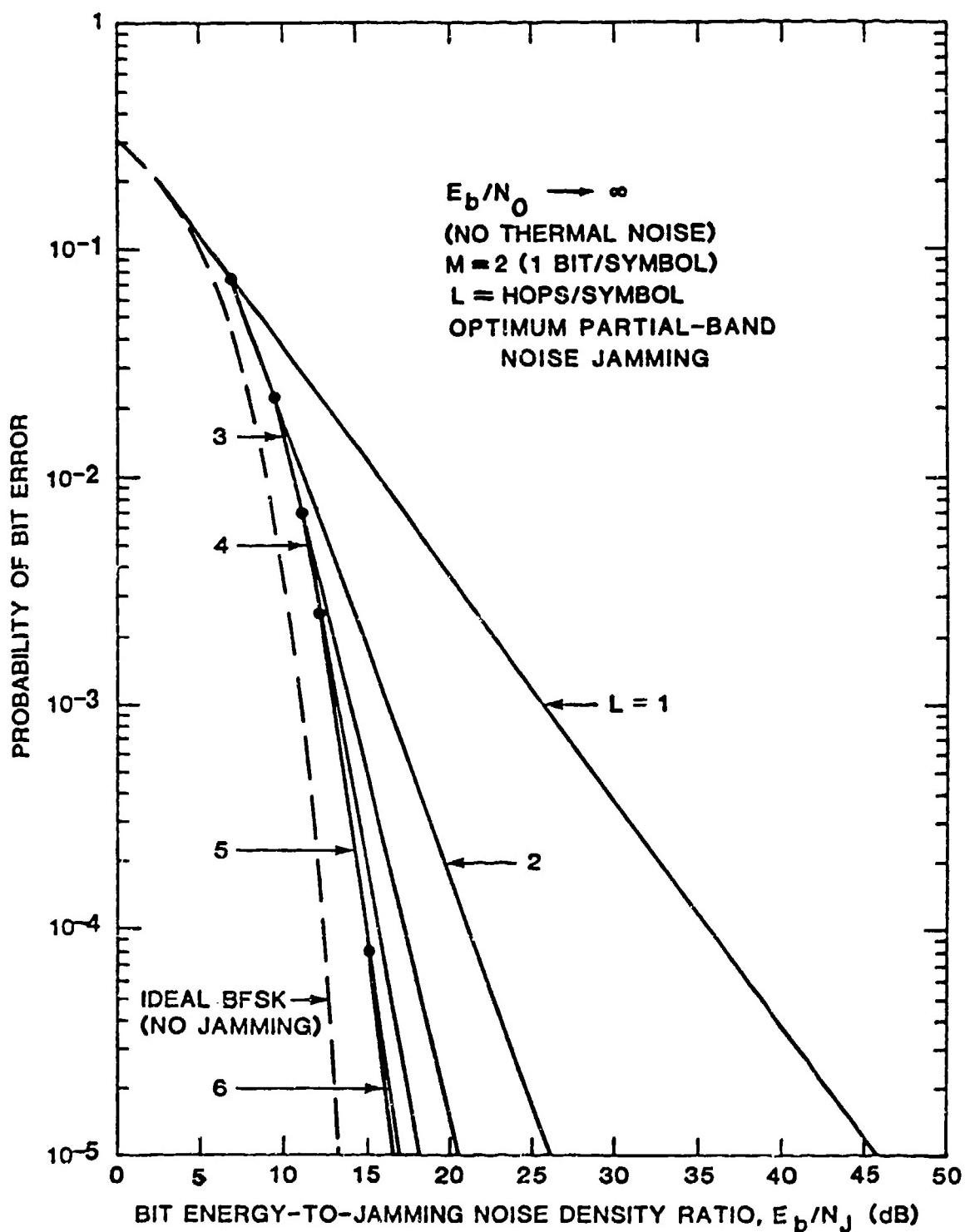


FIGURE 1.2-5 OPTIMUM JAMMING PERFORMANCE OF THE AGC RECEIVER FOR BFSK/FH WITH THE NUMBER OF HOPS/BIT (L) AS A PARAMETER WHEN THERMAL NOISE IS ABSENT (FOR IDEAL BFSK CURVE THE ABSCISSA READS E_b/N_0)

J. S. LEE ASSOCIATES, INC.

Our exact calculations permit the construction of composite curves such as illustrated in Figure 1.2-6, in which the performance of the AGC receiver processing scheme for FH/BFSK ($M=2$) is shown for the optimum L values at different thermal noise levels. It is seen that for $E_b/N_0 > 15$ dB, the use of the proper number of hops per bit enables the communication systems to recover the performance of unjammed BFSK to within 3 dB of SNR. The results for $E_b/N_0 < 15$ dB are very sensitive to thermal noise, and had not been predicted by other workers, who ignored thermal noise.

1.2.2 Impact of Random FH/MFSK (FH/RMFSK) on Analysis

Evaluation of the BER performance of ECCM receiver processing schemes becomes significantly more complex for $M>2$ and $L>1$ when the MFSK symbol frequency assignments are not contiguous but each randomly chosen to be anywhere in the hopping band. The complexity consists in there being many more jamming events than those reflected in equation (1.2-1), since now on each hop there can be from 0 to M of the dehopped symbol frequency slots jammed on a given hop (rather than 0 or M). The probability of bit error expression accordingly must be generalized, giving

$$P(e) = \sum_{\ell_1=0}^L \sum_{\ell_2=0}^L \dots \sum_{\ell_M=0}^L \Pr(\ell_1, \ell_2, \dots, \ell_M) P(e|\ell_1, \ell_2, \dots, \ell_M) \quad (1.2-3)$$

in which the number of jammed hops in each symbol channel is explicitly enumerated and accounted for in the conditional probability of error calculations.

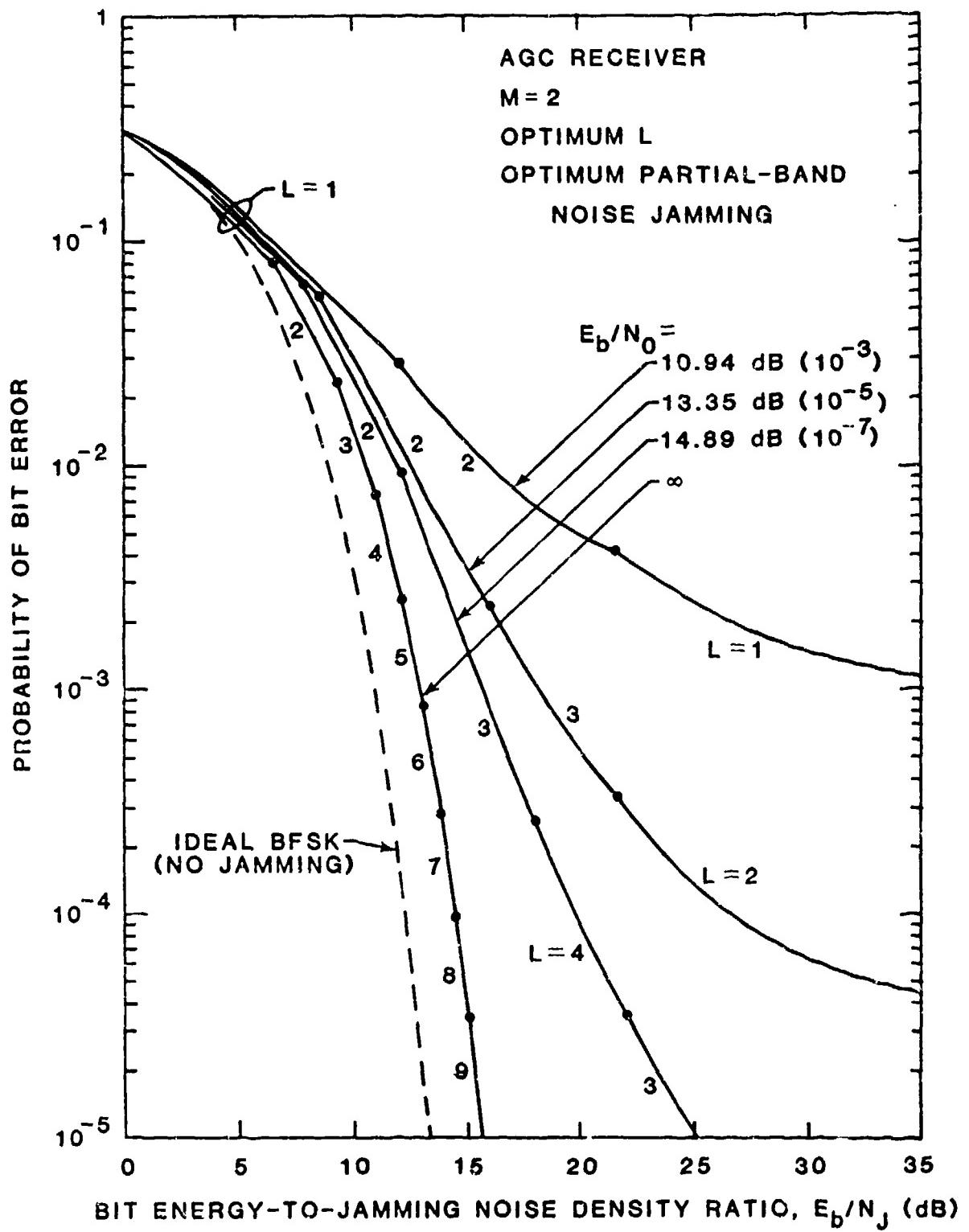


FIGURE 1.2-6 PROBABILITY OF BIT ERROR VS. E_b/N_J FOR AGC FH/BFSK RECEIVER USING OPTIMUM NUMBER (L) OF HOPS/BIT, FOR DIFFERENT VALUES OF E_b/N_0 IN WORST-CASE PARTIAL-BAND NOISE JAMMING

J. S. LEE ASSOCIATES, INC.

From equation (1.2-3) it is apparent that up to $(L + 1)^M$ jamming events may be distinguished, if it can be assumed that the symbol decision is affected only by the total numbers of jammed hops $\{e_m\}$ in the M dehopped channels, rather than by the individual hop patterns. The sheer number of events can therefore become the major factor influencing the magnitude of the receiver effectiveness evaluation task in terms of computational effort. LAI has had experience in the computation of similar expressions in the connection with the evaluation of tone jamming effects on FH/MFSK systems [1].

1.3 SUMMARY OF REPORT

In this section, we will first give a general description of the work. We then summarize the report organization and major findings.

1.3.1 General Description of Work and Approach.

In Sections 1.1 and 1.2 we discussed the fundamental issues concerning ECCM systems and ECCM processing. Now, we treat the more specific ECCM system which we have studied, namely FH/RMFSK in the presence of partial-band noise jamming.

1.3.1.1 Receiver models studied.

A generic soft-decision receiver structure for an FH/RMFSK waveform is shown in Figure 1.3-1. The incoming waveform is dehopped by mixing it separately with M hopping local oscillators controlled by replicas of the M possible hopping sequences available for transmission by the transmitter.

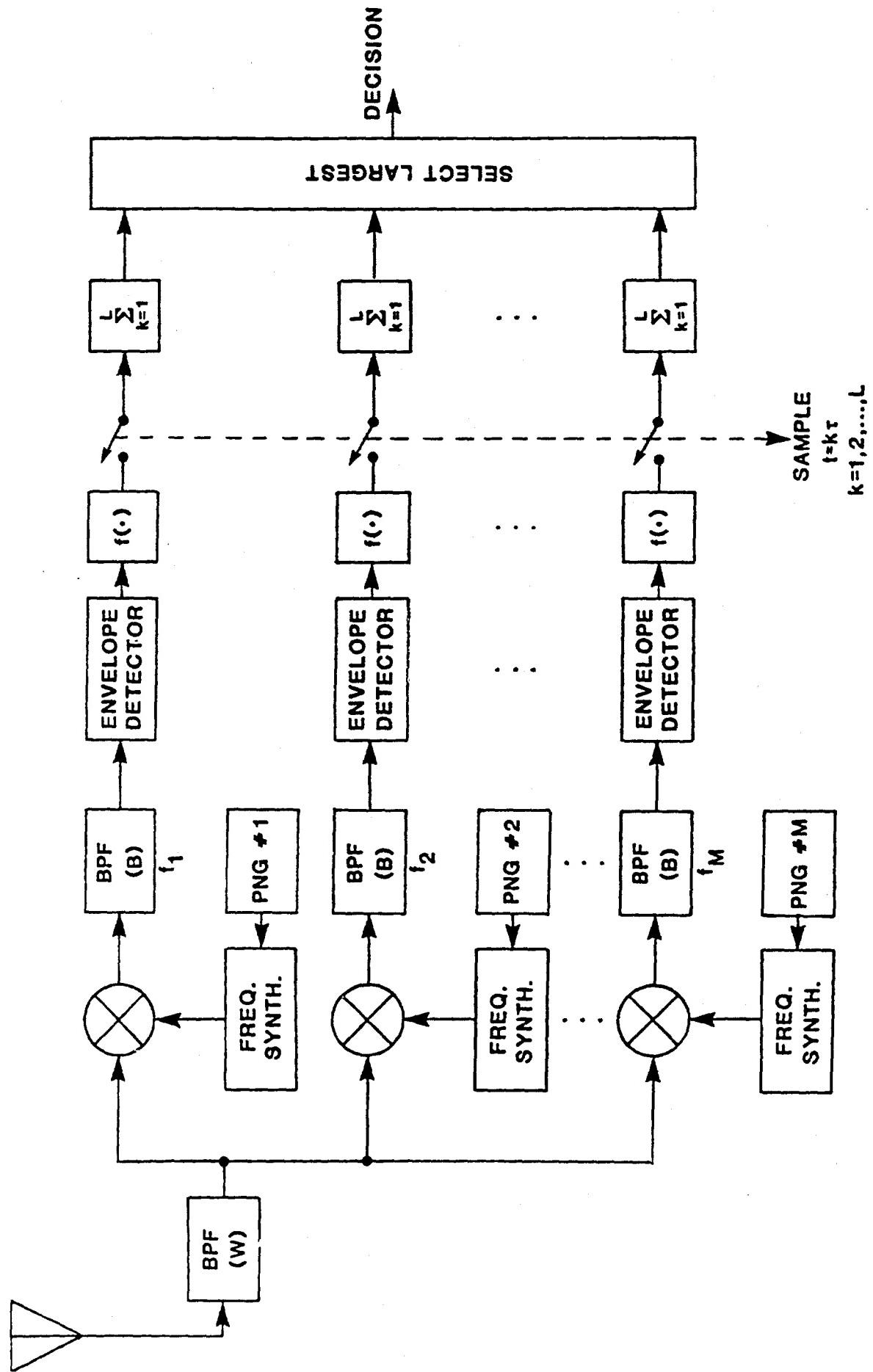


FIGURE 1.3-1 SOFT-DECISION RECEIVER FOR FH/RMFSSK

Thermal noise with power spectral density N_0 is present over the entire bandwidth W . A fraction, γ , of the band is jammed by bandlimited white Gaussian noise of power spectral density N_J/γ , where $N_J \triangleq J/W$ with J being the total jammer power. The jamming fraction, γ , is constrained to the range $0 < \gamma \leq 1$.

The relation between the jammed bandwidth γW and the FH/RMFSK waveform is illustrated in Figure 1.3-2. On any given hop, anywhere from 0 to M of the possible signalling frequencies may have hopped into the jammed portion of the band; thus a multitude of jamming events may occur. Let the L hops for a given symbol be referred to individually by the index k ($k = 1, 2, \dots, L$). The jamming events for the k th hop can be described in terms of which of the M symbol frequencies are jammed, and which are not. In general there are 2^M possibilities for a given hop, which we may specify by the indicator vector

$$\underline{v}_k = (v_{1k}, v_{2k}, \dots, v_{Mk}) \quad (1.3-1)$$

where

$$v_{mk} = \begin{cases} 1 & \text{if symbol slot } m \text{ is jammed on hop } k \\ 0 & \text{if not;} \end{cases} \quad m = 1, 2, \dots, M; \quad k = 1, 2, \dots, L. \quad (1.3-2)$$

For the L hops comprising a symbol, there are 2^{ML} possible jamming events, and these can be specified individually by the $M \times L$ indicator matrix $[v] \equiv [v_{mk}]$.

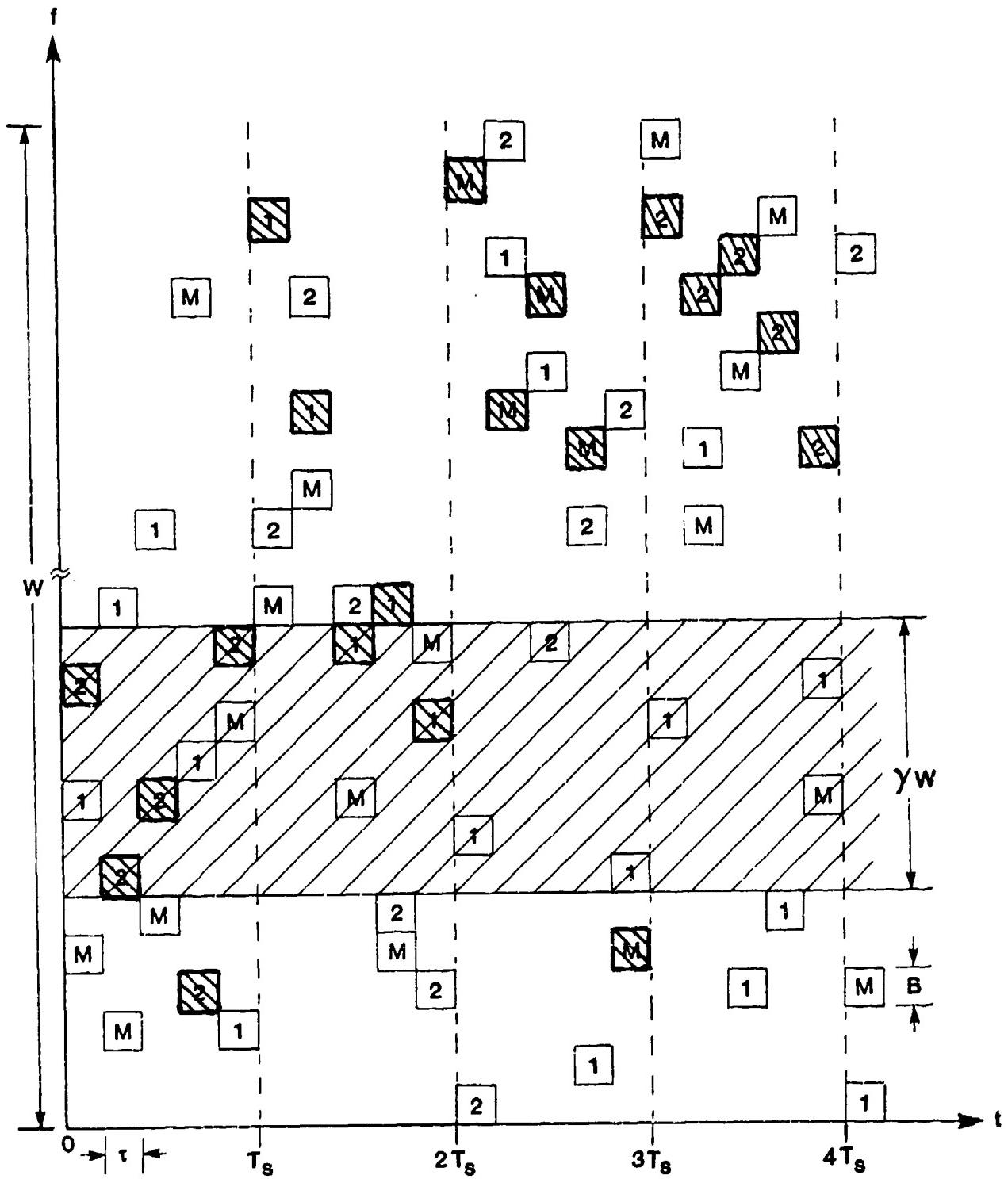


FIGURE 1.3-2 PARTIAL-BAND JAMMING OF FH/RMFSK

J. S. LEE ASSOCIATES, INC.

Each dehopped channel, corresponding to one of the M possible symbols, is then passed through a bandpass filter of width B Hz and the filter output is envelope detected. The output of each linear envelope detector is subjected to a function $f(\cdot)$; the form of this function defines the particular receiver structure.* Table 1.3-1 gives the forms of $f(\cdot)$ for the several receiver structures we include in the study. The modified envelopes are sampled once per hop and the samples in each channel are summed over the L hops comprising a symbol. The largest of these sums is selected and the index identifying the channel in which it occurred is outputted as the symbol decision.

As an alternative to the soft-decision receiver scheme described above, we may also consider the hard-decision receiver structure which is shown in Figure 1.3-3. The processing in this hard-decision receiver is identical with the soft-decision receiver up to the outputs of the samplers. In the hard-decision receiver, unlike the soft-decision receiver, the samples are not summed; rather, a symbol decision is made each hopping interval, giving a sequence of L decisions. These L decisions may be considered as a noise-corrupted received code-word in an M -ary repetition code wherein the transmitted symbol is repeated L times; thus the sequence is fed into an L -hop M -ary repetition code decoder which delivers the final decision as to which symbol was transmitted.

1.3.1.2 Jamming model and measure of effectiveness .

The partial-band noise jamming model was shown in Figure 1.1-2.

*As long as $f(\cdot)$ is a memoryless transformation, the order of applying $f(\cdot)$ and the sampling may be interchanged without altering the receiver's performance.

TABLE 1.3-1
RECEIVER PROCESSING FUNCTIONS STUDIED

RECEIVER TYPE	$f(\cdot)$
Square-Law Linear Combining	$f(x_i) = x_i^2$
Square-Law with Clipper	$f(x_i) = \begin{cases} x_i^2, & x_i < \sqrt{\eta} \\ \eta, & x_i \geq \sqrt{\eta} \end{cases}$
Square-Law AGC	$f(x_i) = x_i^2 / \sigma_i^2$
Self-normalizing receiver	$f(x_i) = \frac{x_i^2}{\sum_{j=1}^M x_j^2}$

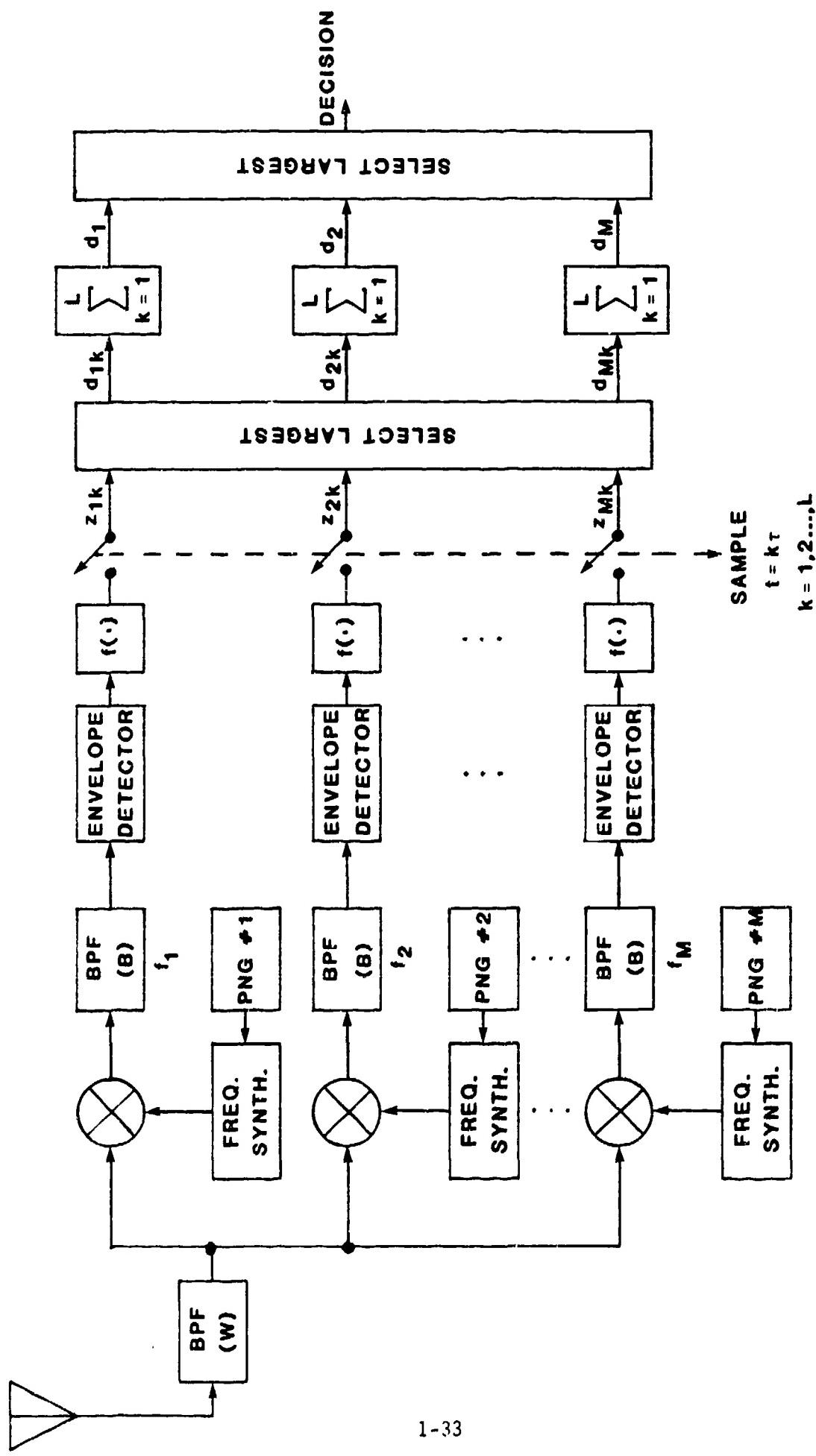


FIGURE 1.3-3 HARD-DECISION RECEIVER FOR FH/RMFSK

The measure of the effectiveness of the jammer is the degradation of the communicator's bit error probability inflicted by the presence of the jamming. Since the bit error probability $P_b(e)$ will depend upon the jamming event, we must average the error probability over the jamming events. Thus, the measure of effectiveness of the jamming is

$$P_b(e; E_b/N_0, E_b/N_J, \gamma, M, L) = \sum_{[v]} P_b(e; E_b/N_0, E_b/N_J, \gamma, M, L| [v]) \pi_L[v]$$

where $\pi_L[v]$ is the probability of jamming event $[v]$ occurring over the L hops of the M -ary symbol. Thus the required analysis may be divided into two parts: determination of $\pi_L[v]$ and determination of $P_b(e; E_b/N_0, E_b/N_J, \gamma, M, L| [v])$. These two parts can then be combined to perform the final optimization, namely finding the receiver performance under the optimum jamming fraction γ , $\max_{\gamma} P_b(e; \gamma)$.

1.3.1.3 Organization of report.

In Section 2 we address parts of the analysis considered preliminary or containing aspects common to the several receiver types. This material includes enumeration of jamming events and analysis of their probabilities, as well as an analysis of the hard-decision receiver.

Sections 3 to 6 are devoted, respectively, to analysis and numerical results for the worst-case partial-band noise jamming error performances of FH/RMFSK using the linear combining receiver, the adaptive gain control (AGC) type receivers, the clipper receiver, and the self-normalizing receiver.

Section 7 first provides analysis and results for the performance of FH/RMFSK in follow-on noise jamming, then comparisons of RMFSK receivers

J. S. LEE ASSOCIATES, INC.

with regard to their overall relative error performance, their performances in both RMFSK and MFSK, and their success in using diversity (multiple hops per symbol) to mitigate jamming effects.

Section 8 considers issues related to implementation of the FH/RMFSK receivers, including a discussion of possible measurement approaches to support the ECCM weighting schemes, and an assessment of the effect of using practical measurements (instead of a priori information) on the system performance. Conclusions and recommendations growing out of our study are included in Section 8 also.

1.3.2 Summary of Findings.

Here we only briefly cite the more significant findings from our study; more detailed information is contained throughout the report.

The overall significance of the work we have accomplished may be described as follows: For the first time, the expected performance of an FH/RMFSK system using L hops per symbol and soft decisions, in both thermal noise and worst-case partial-band jamming noise, has been derived and calculated. Moreover, we have demonstrated through direct analysis and calculation of bit error rate (BER) that the performances of certain practical soft-decision ECCM receivers (using no a priori or side information) are quite acceptable, being very close to those for idealized receivers (using a priori information on received noise and jamming conditions). While we have shown that the hard-decision receiver does implement a form of ECCM processing (not previously shown) against PBNJ in the most simple manner, it cannot be

J. S. LEE ASSOCIATES, INC.

considered a viable alternative unless the system's unjammed SNR is quite high.

Specifically, we find that:

(a) Generally random frequency hopping MFSK is more vulnerable to partial-band noise jamming than is conventional FH/MFSK for $M > 2$ and $L > 1$. However, for certain diversity weighting schemes the increased vulnerability is small enough to justify saying that the two hopping systems achieve comparable performance for $M = 2$ or 4. For one combining scheme studied, the self-normalizing receiver, FH/RMFSK performs better than FH/MFSK for $M = 2$.

(b) A diversity effect for L hops/symbol is observed for RMFSK using nonlinear hop combining, in the same manner as for MFSK and subject to the same condition that thermal noise is relatively small.

(c) Using optimum diversity values, if thermal noise is negligible ($E_b/N_0 \geq 20$ dB), FH/RMFSK with ideal nonlinear combining can exhibit a nearly exponential dependence upon E_b/N_j , as opposed to an inverse linear one for no diversity; the jamming then is limited to inflicting about a 4 dB loss in system performance. However, this effect is very sensitive to the amount of thermal noise present, since the jammed BER cannot be better than the unjammed error, and the use of diversity tends to degrade the no-jamming performance due to noncoherent combining losses.

(d) Simple nonlinear combining receivers can duplicate the ideal receiver optimum diversity performance with about a one-dB loss when $E_b/N_0 > E_b/N_j$; the hard decision receiver can approach to within 2 dB with a sufficiently high E_b/N_0 .

J. S. LEE ASSOCIATES, INC.

2.0 PRELIMINARY ANALYSES

The general approach to be followed in obtaining the probability of error for the multi-hops/symbol FH/RMFSK communications system under partial band noise jamming is to expand the total probability of error in terms of individual jamming events:

$$\begin{aligned} P(e) &= \sum_{\text{jamming events}} P(e, \text{jamming event}) \\ &= \sum_{\text{jamming events}} \Pr(\text{jamming event}) P(e|\text{jamming event}), \end{aligned}$$

where $P(e|\text{jamming event})$ is the probability of error conditioned upon the occurrence of a particular jamming event. In Section 2.1, we consider a general formulation for this conditional probability, and in Section 2.2 the jamming events and their probabilities are developed. The computational procedures necessary for efficient evaluation of the error probability are discussed in Section 2.3. These analyses and procedures are applied to specific receiver structures beginning in Section 3.

2.1 CONDITIONAL PROBABILITY OF ERROR

The generic form of the receiver to be analyzed for reception of FH/RMFSK is shown in Figure 2.1-1. In effect it is an M-channel receiver with IF frequencies f_1, f_2, \dots, f_M ; M pseudorandom sequence generators, assumed to be in synchronism with the transmitter, command the frequency synthesizers used to tune the M channels to B-Hz wide frequency slots, one of which will be occupied by signal energy on a given hop. The message information is

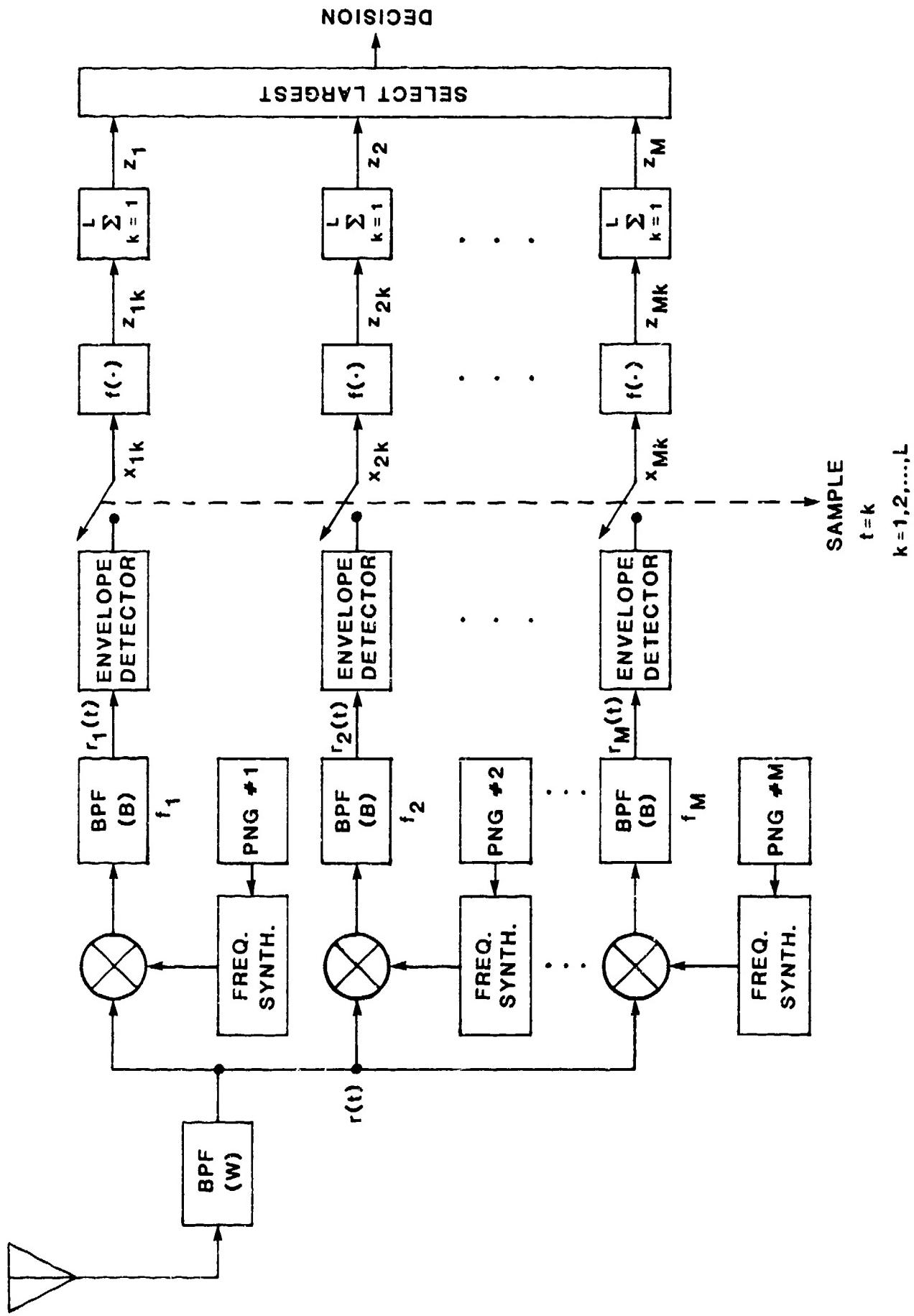


FIGURE 2.1-1 GENERIC RECEIVER FOR FH/RMFSK

J. S. LEE ASSOCIATES, INC.

conveyed through selection of the (randomly-shifted) IF (symbol) frequency for transmission of signal energy. In order to determine at which symbol frequency the signal is present, each of the IF waveforms $r_1(t), r_2(t), \dots, r_M(t)$ is first subjected to envelope detection, then sampled before processing through memoryless, possibly nonlinear devices with transfer functions $f(\cdot)$. The outputs of these devices are the per-hop decision statistics $\{z_{mk}\}$, which are accumulated to form the final decision statistics

$$z_m = \sum_{k=1}^L z_{mk}, \quad m = 1, 2, \dots, M. \quad (2.1-1)$$

In the following subsections, we formulate the probability of error associated with the decision performed by the FH/RMFSK receiver, conditioned upon the possible jamming events.

2.1.1 Assumed Signals, Noise and Jamming.

After dehopping, the received signal is assumed equally likely to be present in any one of the M channels for the entire symbol period $T_s = L\tau$, where τ is the hop period and L is the number of hops per MFSK symbol. Without loss of generality, we assume that the signal with power S is in channel 1, or

$$s(t) = \sqrt{2S} \cos(\omega_1 t + \varphi_k), \quad (k-1)\tau < t \leq k\tau, \quad k = 1, 2, \dots, L, \quad (2.1-2)$$

where φ_k is an arbitrary carrier phase and $\omega_1 = 2\pi f_1$.

Thermal noise is considered also to be present in each channel, and is assumed to be zero-mean narrowband Gaussian noise with variance $\sigma_N^2 = N_0 B$, where $N_0/2$ is the (two-sided) noise power spectral density and B is the bandwidth of each channel. Thus for no jamming the samples of the M envelope detector outputs on the k th hop are the variables

J. S. LEE ASSOCIATES, INC.

$$x_{1k} = \left[(\sqrt{2S} \cos \theta_k + n_{c1k})^2 + (\sqrt{2S} \sin \theta_k + n_{s1k})^2 \right]^{\frac{1}{2}} \quad (2.1-3a)$$

and

$$x_{mk} = \left(n_{cmk}^2 + n_{smk}^2 \right)^{\frac{1}{2}}, \quad m = 2, 3, \dots, M, \quad (2.1-3b)$$

where n_{cmk} , n_{smk} , $m = 1, 2, \dots, M$; $k = 1, 2, \dots, L$, are the independent noise quadrature components in the channels at the sample times $t_k = k\tau$, with

$$E(n_{cmk}^2) = E(n_{smk}^2) = \sigma_N^2 = N_0 B, \text{ for all } m, k. \quad (2.1-4)$$

Because the MFSK symbol slots are hopped independently, none, some, or all of the dehopped channels can be jammed on an individual hop. The possible combinations of such events and their probabilities are discussed in Section 2.2.

When jamming noise is present in a channel, it is assumed to be zero-mean narrowband Gaussian noise with variance $\sigma_j^2 = N_j B / \gamma$, where $N_j / 2$ is the (two-sided) noise power spectral density averaged over the system bandwidth; and γ is the fraction of this bandwidth which is jammed. That is,

$$N_j = \frac{J}{W}, \quad (2.1-5)$$

where J is the total jammer power and W is the system bandwidth. When the channels are jammed on the k -th hop, the combination of jamming and thermal noise produces the detector output samples

$$x_{1k} = \left[(\sqrt{2S} \cos \theta_k + n_{c1k} + j_{c1k})^2 + (\sqrt{2S} \sin \theta_k + n_{s1k} + j_{s1k})^2 \right]^{\frac{1}{2}} \quad (2.1-6a)$$

$$x_{mk} = \left[(n_{cmk} + j_{cmk})^2 + (n_{smk} + j_{smk})^2 \right]^{\frac{1}{2}}, \quad m = 2, 3, \dots, M, \quad (2.1-6b)$$

where j_{cmk} , j_{smk} , $i = 1, 2, \dots, M$; $k = 1, 2, \dots, L$, are the independent jamming noise quadrature components in the channels at the sample times, with

J. S. LEE ASSOCIATES, INC.

$$E(j_{cmk}^2) = E(j_{smk}^2) = \sigma_J^2 = N_J B / \gamma, \text{ for all } m, k, \quad (2.1-7)$$

and γ is the fraction of the system bandwidth which is jammed.

In a summary way, we can express the detector output samples by

$$x_{1k} = \sigma_{1k} \left[\left(\sqrt{\frac{2S}{\sigma_{1k}^2}} \cos \theta_k + u_{c1k} \right)^2 + \left(\sqrt{\frac{2S}{\sigma_{1k}^2}} \sin \theta_k + u_{s1k} \right)^2 \right]^{\frac{1}{2}} \quad (2.1-8a)$$

$$x_{mk} = \sigma_{mk} (u_{cmk}^2 + u_{smk}^2)^{\frac{1}{2}}, \quad m = 2, 3, \dots, M, \quad (2.1-8b)$$

where u_{cmk} and u_{smk} are independent, unit-variance, zero-mean Gaussian random variables, and for channel m on hop k ,

$$\sigma_{mk}^2 = \begin{cases} \sigma_N^2 = N_0 B, & \text{unjammed} \\ \sigma_T^2 = \sigma_N^2 + \sigma_J^2 = (N_0 + N_J/\gamma) B, & \text{jammed} \end{cases} \quad (2.1-9a)$$

or, more compactly,

$$\sigma_{mk}^2 = \sigma_N^2 + v_{mk} \sigma_J^2. \quad (2.1-9b)$$

In this last equation $v_{mk} = 1$ if channel m is jammed on hop k , and $v_{mk} = 0$ if not. Thus x_1 is σ_{1k} times a Rician random variable with SNR

$$\rho_k = S / \sigma_{1k}^2, \quad (2.1-10)$$

and x_{mk} , $m > 1$, is σ_{mk} times a Rayleigh random variable.

2.1.2 Conditional Error Probability Formulation

Assuming equally likely M -ary symbols, we may write the conditional symbol error probability as

J. S. LEE ASSOCIATES, INC.

$$P_s(e|[\nu]) = P_s(e|[\nu], m_1 \text{ transmitted}) \quad (2.1-11)$$

in which $[\nu]$ is a matrix describing the jamming event for L hops, with elements ν_{mk} .

For M a power of two ($M=2^K$), the conditional bit error probability is obtained from the conditional symbol error probability using the relation

$$P_b(e|[\nu]) = \frac{M/2}{M-1} P_s(e|[\nu]). \quad (2.1-12)$$

Since for $M > 2$ there are many error events but only one correct decision, it is convenient to write the conditional symbol error probability in terms of the probability of a correct decision as

$$\begin{aligned} P_s(e|[\nu]) &= 1 - P_s(c|m_1, [\nu]) \\ &= 1 - \Pr\{z_2 < z_1, z_3 < z_1, \dots, z_M < z_1\}. \end{aligned} \quad (2.1-13)$$

In terms of the pdf's for the statistics, this becomes

$$P_s(e|[\nu]) = 1 - \int_0^\infty d\beta_1 \int_0^{\beta_1} d\beta_2 \dots \int_0^{\beta_1} d\beta_M p_z(\beta_1, \beta_2, \dots, \beta_M | [\nu]); \quad (2.1-14)$$

if the decision statistics are independent, then

$$P_s(e|[\nu]) = 1 - \int_0^\infty d\beta p_{z_1}(\beta | [\nu]) \prod_{m=2}^M \int_0^\beta d\alpha_m p_{z_m}(\alpha_m | [\nu]). \quad (2.1-15)$$

For certain receiver structures, the probability distributions of the individual channel statistics $\{z_m\}$ are mutually independent. This relationship causes the conditional probability of error to depend only on the number of hops jammed in each of the M channels, rather than on specific patterns, and greatly reduces the number of distinguishable jamming events.

J. S. LEE ASSOCIATES, INC.

2.2 ENUMERATION OF DISTINGUISHABLE JAMMING EVENTS

Examination of the conditional error probability expression reveals that the same conditional error will occur for several different values of the fundamental jamming event matrix, $[v]$. Therefore, in terms of error probability, there is a number of distinguishable jamming events which is smaller than the 2^{ML} possible values of the $[v]$ matrix. It is important to identify and enumerate these distinguishable jamming events in order to take advantage of the savings in computation which will result.

In this section we first identify and enumerate the distinguishable jamming events, then investigate methods for calculating their probabilities.

2.2.1 Definition of Distinguishable Jamming Events.

The conditional error probability, as shown in Section 2.1, is a function of given values of the $[v]$ matrix elements v_{mk} , where $m=1$ to M (the number of MFSK channels) and $k=1$ to L (the number of hops per symbol). Often this function can be written

$$P(e|[v]) = P(e|v_{11}, v_{12}, \dots, v_{1L}; \dots; v_{M1}, \dots, v_{ML}) \\ = f\left(\sum_{k=1}^L v_{1k}, \sum_{k=1}^L v_{2k}, \dots, \sum_{k=1}^L v_{mk}\right). \quad (2.2-1)$$

Thus, if we define the row sums

$$r_m \triangleq \sum_{k=1}^L v_{mk}, \quad (2.2-2)$$

J. S. LEE ASSOCIATES, INC.

the conditional $P(e)$ is a function only of these sums*, which are to be interpreted as the number of hops jammed in the respective channels. This fact can be expressed by the relation

$$\begin{aligned} P(e|v) &= f(\underline{\ell}_1, \underline{\ell}_2, \underline{\ell}_3, \dots, \underline{\ell}_M) \\ &\equiv f(\underline{\ell}), \end{aligned} \quad (2.2-3)$$

where $\underline{\ell}$ is the vector of $\underline{\ell}_m$ components.

Since each $\underline{\ell}_m$ can take integer values from 0 to L , there $(L+1)^M$ possible jamming events described by the vector $\underline{\ell}$. This is a considerable savings in numbers of jamming events, as illustrated by Table 2.2-1.

TABLE 2.2-1
NUMBER OF JAMMING EVENTS

M	L	#{{v} events, 2^{ML} }	#{{ $\underline{\ell}$ events, $(L+1)^M$ }
2	1	4	4
	2	16	9
	3	64	16
	4	256	25
4	1	16	16
	2	256	81
	3	4,096	256
	4	65,536	625
8	1	256	256
	2	65,536	6,561
	3	16,777,216	65,536
	4	4,294,967,296	390,625

2.2.2 Smallest Set of Distinguishable Jamming Events.

A further reduction in the number of distinguishable jamming events

*An exception to this condition results for an ECCM processing scheme studied in Section 4.

J. S. LEE ASSOCIATES, INC.

results from noticing that permutation of the non-signal channel quantities (ℓ_2 to ℓ_M) does not affect the conditional $P(e)$. That is,

$$\begin{aligned}
 f(\underline{\ell}) &= f(\ell_1, \ell_2, \ell_3, \dots, \ell_M) \\
 &= f(\ell_1, \ell_M, \ell_3, \ell_4, \dots, \ell_2) \\
 &= f(\ell_1, \ell_5, \ell_{M-1}, \ell_2, \dots, \ell_3) \\
 &= \dots \text{etc.}
 \end{aligned} \tag{2.2-4}$$

Thus we can restrict our attention to just one permutation of the set of values $\underline{\ell} = (\ell_1, \ell_2, \dots, \ell_M)$. A convenient way to represent the permutations is the ordered set of numbers

$$\underline{\ell}' \triangleq \{\ell_1, \ell_2, \ell_3, \dots, \ell_M : \ell_2 \leq \ell_3 \leq \dots \leq \ell_M\}. \tag{2.2-5}$$

There are, from Appendix B.3.,

$$\sum_{\ell_1=0}^L \sum_{M=0}^M \sum_{\ell_{M-1}=0}^{\ell_M} \dots \sum_{\ell_3=0}^{\ell_4} \sum_{\ell_2=0}^{\ell_3} = (L+1) \binom{M-1+L}{M-1} \tag{2.2-6}$$

such ordered $\underline{\ell}'$ vectors, which represent the minimum number of distinguishable events. Example values are given in Table 2.2-2.

TABLE 2.2-2
MINIMUM NUMBER OF DISTINGUISHABLE EVENTS

<u>M</u>	<u>L</u>	<u>#$\{\underline{\ell}'$ events</u>
2	1	4
	2	9
	3	16
	4	25
4	1	8
	2	30
	3	80
	4	175
8	1	16
	2	108
	3	480
	4	1,650

Each of the distinguishable jamming events represents a certain number of events with identical jamming effects. The number of $\underline{\ell}$ vectors thereby represented by a particular ordered vector $\underline{\ell}'$ is

$$\#(\underline{\ell} \leftrightarrow \underline{\ell}') = \binom{M-1}{n_0, n_1, \dots, n_L} \quad (2.2-7a)$$

$$= \frac{(M-1)!}{n_0! n_1! \dots n_L!} \quad (2.2-7b)$$

where

n_ℓ = number of ℓ_m which equal ℓ ; $\ell=0,1,\dots,L$; $m > 1$

and we have

$$\sum_{\ell=0}^L n_\ell = M-1. \quad (2.2-7d)$$

For example, for $M=8$ and $L=6$, the number of jamming event vectors $\underline{\ell}$ represented by the ordered vector $\underline{\ell}' = (\ell_1; 0,0,2,3,3,4,5)$ is

$$\binom{7}{2,0,1,2,1,1,0} = \frac{7!}{2!0!1!2!1!1!0!} = 1260. \quad (2.2-8)$$

As a check on this enumeration, we find that the total number of $\underline{\ell}$ jamming events is given by

$$\begin{aligned} \#(\underline{\ell}) &= \sum_{\ell_1=0}^L \sum_{\ell_M=0}^L \sum_{\ell_{M-1}=0}^{\ell_M} \sum_{\ell_3=0}^{\ell_{M-1}} \sum_{\ell_2=0}^{\ell_3} \binom{M-1}{n_0, n_1, \dots, n_L} \\ &= (L+1) \sum_{\ell_M=0}^L \dots \sum_{\ell_2=0}^{\ell_3} \binom{M-1}{n_0, n_1, \dots, n_L}. \end{aligned} \quad (2.2-9)$$

It can be shown (see Appendix B) that the summation in (2.2-9) is equal to $(L+1)^{M-1}$. Thus the total number of $\underline{\lambda}$ vectors computed by (2.2-9) is $(L+1)^M$, which agrees with our previous enumeration.

2.2.3 Jamming Event Probabilities, Single Hop.

Given a jamming event described by the vector $\underline{\lambda}$, what is the probability of the event under the random hopping scheme and partial-band noise jamming? To find this answer, we first consider the case of one hop per MFSK symbol, or $L=1$.

The jammer spectrum is assumed to be flat, with one-sided power spectral density $J/\gamma W$, where γ is the fraction of the system bandwidth occupied by the jammer. There are $N=W/B$ possible symbol frequency slots, and it is assumed that M of these slots are assigned randomly to the MFSK symbol on each hop. At the same time the number of slots containing jamming power is

$$q \triangleq \gamma N, \quad (2.2-10)$$

assumed to be an integer. That is, $\gamma = q/N$, with q an integer.

The probability that n of the M symbol slots are jammed on a given hop is

$$\begin{aligned} \pi_n &= \frac{q}{N} \cdot \frac{q-1}{N-1} \cdots \frac{q-n+1}{N-n+1} \cdot \frac{N-q}{N-n} \cdots \frac{N-q-M+n+1}{N-M+1} \\ &= \frac{\binom{N-M}{q-n}}{\binom{N}{q}}, \quad n=0,1,2,\dots,\min(q,M). \end{aligned} \quad (2.2-11)$$

Note that the probability π_n is valid for several different jamming events since

$$n_k = \sum_{m=1}^M v_{mk} \quad (2.2-12)$$

on the k-th hop. In fact there are $\binom{M}{n}$ jamming events for $L=1$ which have probability π_n . Thus

$$\sum_{n=0}^L \binom{M}{n} \pi_n = 1, \quad (2.2-13)$$

as required.

In terms of distinguishable jamming events, we differentiate between whether the signal channel is jammed ($v_{1k} = 1$) or not ($v_{1k} = 0$), and describe single-hop jamming events by the pair of numbers (v_{1k}, r_k) with

$$r_k \triangleq \sum_{m=2}^M v_{mk}, \quad (2.2-14)$$

the number of non-signal channels jammed. We have

$$\Pr\{v_{1k}, r_k\} = \pi(v_{1k} + r_k) \equiv \pi_n, \quad n = v_{1k} + r_k \quad (2.2-15)$$

and

$$\sum_{v_{1k}=0}^1 \sum_{r_k=0}^{M-1} \binom{M-1}{r_k} \pi(v_{1k} + r_k) = 1. \quad (2.2-16)$$

2.2.4 Jamming Event Probabilities For Multiple Hops, Characteristic Function Method.

For $L > 1$ hops per symbol, as discussed in Sections 2.2.1 and 2.2.2, the distinguishable jamming events are described by the vectors $\underline{\ell} = (\ell_1, \ell_2, \dots, \ell_M)$. We note that

$$\underline{\ell} = \sum_{k=1}^L \underline{v}_k \quad (2.2-17)$$

where $\underline{v}_k = (v_{1k}, v_{2k}, \dots, v_{Mk})$ is the vector whose elements are the k -th column of the matrix $[v]$ of fundamental jamming events. Since the hopping pattern is assumed to be independent from hop to hop, we may treat $\underline{\ell}$ as the sum of identically distributed discrete random vectors, and find the probabilities of the $\underline{\ell}$ jamming events from the characteristic function of the \underline{v} jamming events.

The characteristic function of any one of the random vectors \underline{v}_k is given by

$$\begin{aligned} \phi_{\underline{v}_k}(\underline{\mu}; M) &= E\{\exp[j\underline{v}_k \cdot \underline{\mu}]\} \\ &= E\{\exp[j\mu_1 v_{1k} + j\mu_2 v_{2k} + \dots + j\mu_M v_{Mk}]\} \\ &= \pi_0 + \pi_1 (e^{j\mu_1} + e^{j\mu_2} + \dots + e^{j\mu_M}) \\ &\quad + \pi_2 (e^{j\mu_1 + j\mu_2} + \dots + e^{j\mu_{M-1} + j\mu_M}) \\ &\quad + \dots \\ &\quad + \pi_M e^{j\mu_1 + j\mu_2 + \dots + j\mu_M}. \end{aligned} \quad (2.2-18)$$

J. S. LEE ASSOCIATES, INC.

For example, for $M=2$ and 4 the characteristic functions for one hop are

$$\phi_{\underline{v}}(\underline{\mu};2) = \pi_0 + \pi_1 e^{j\mu_1} + \pi_1 e^{j\mu_2} + \pi_2 e^{j\mu_1+j\mu_2} \quad (2.2-19)$$

and

$$\begin{aligned} \phi_{\underline{v}}(\underline{\mu};4) = & \pi_0 + \pi_1 e^{j\mu_1} + \pi_1 e^{j\mu_2} + \pi_1 e^{j\mu_3} + \pi_1 e^{j\mu_4} \\ & + \pi_2 e^{j\mu_1+j\mu_2} + \pi_2 e^{j\mu_1+j\mu_3} + \pi_2 e^{j\mu_1+j\mu_4} \\ & + \pi_2 e^{j\mu_2+j\mu_3} + \pi_2 e^{j\mu_2+j\mu_4} + \pi_2 e^{j\mu_3+j\mu_4} \\ & + \pi_3 e^{j\mu_1+j\mu_2+j\mu_3} + \pi_3 e^{j\mu_1+j\mu_2+j\mu_4} \\ & + \pi_3 e^{j\mu_1+j\mu_3+j\mu_4} + \pi_3 e^{j\mu_2+j\mu_3+j\mu_4} \\ & + \pi_4 e^{j\mu_1+j\mu_2+j\mu_3+j\mu_4} . \end{aligned} \quad (2.2-20)$$

In this characteristic function, there are 2^M terms, one for each of the events described by \underline{v}_k .

The characteristic function for $\underline{\ell}$ is simply that of \underline{v}_k , raised to the L -th power:

$$\phi_{\underline{\ell}}(\underline{\mu};M) = [\phi_{\underline{v}}(\underline{\mu};M)]^L . \quad (2.2-21)$$

For example, for $M=2$,

$$\begin{aligned} \phi_{\underline{\ell}}(\underline{\mu};2) &= [\phi_{\underline{v}}(\underline{\mu};2)]^L \\ &= \sum_{n_0, n_1, n_2, n_3}^L \binom{L}{n_0, n_1, n_2, n_3} \pi_0^{n_0} \pi_1^{n_1} \pi_2^{n_2} \pi_2^{n_3} \\ &\times \exp\{j(n_1+n_3)\mu_1 + j(n_2+n_3)\mu_2\}, \end{aligned} \quad (2.2-22a)$$

J. S. LEE ASSOCIATES, INC.

where the summation is over all (n_0, n_1, n_2, n_3) such that

$$\sum_{q=0}^{2^M-1} n_q = L . \quad (2.2-22b)$$

The discrete probability density function (pdf) for $\underline{\ell}$ is the inverse Fourier transform of $\phi_{\underline{\ell}}(\underline{u}; M)$. Again, for $M=2$, the pdf $p(\underline{\ell}; M, L)$ for $\underline{\ell}$ is

$$p(\underline{\ell}; 2, L) = \sum_{n_0, n_1, n_2, n_3} \binom{L}{n_0, n_1, n_2, n_3} \pi_0^{n_0} \pi_1^{n_1+n_2} \pi_2^{n_3} \\ \times \delta(n_1+n_3-\ell_1) \delta(n_2+n_3-\ell_2), \quad (2.2-23)$$

which can be used to find the individual $\underline{\ell}$ vector probabilities

$$\Pr\{\underline{\ell}; 2, L\} = \sum_{n=0}^L \binom{L}{n, L-\ell_2-n, L-\ell_1-n, \ell_1+\ell_2+n-L} \\ \times \pi_0^n \pi_1^{2L-\ell_1-\ell_2-2n} \pi_2^{\ell_1+\ell_2-L+n} . \quad (2.2-24)$$

In (2.2-24) it is realized that the combinatorial factor is zero if any of its parameters is negative. As an example for $L=3$ and $M=2$,

$$\Pr\{\ell_1 = 1, \ell_2 = 2; 2, 3\} \\ = \binom{3}{0, 1, 2, 0} \pi_1^3 + \binom{3}{1, 0, 1, 1} \pi_0 \pi_1 \pi_2 \\ = 3\pi_1^3 + 6\pi_0 \pi_1 \pi_2 . \quad (2.2-25)$$

J. S. LEE ASSOCIATES, INC.

An aid to checking our calculations is the fact that each $\underline{\lambda}$ event corresponds to

$$\binom{L}{\lambda_1} \binom{L}{\lambda_2} \dots \binom{L}{\lambda_M} \quad (2.2-26)$$

[v] matrix events. In (2.2-25) therefore, there are $\binom{3}{1} \binom{3}{2} = 9$ terms.

A complete table of $M=2$ jamming event probabilities for $L=1$ to 4 is given in Table 2.2-3, and an equation for computing these probabilities for $M=4$ is included as Table 2.2-4.

2.2.5 Jamming Event Probabilities For Multiple Hops, Convolution Method.

Since $\underline{\lambda}$ is the sum of L independent random event vectors \underline{v}_k , the pdf for $\underline{\lambda}$ is the L -fold convolution of the pdf for \underline{v}_k :

$$p(\underline{\lambda}; M, L) = p(\underline{v}_1; M) * p(\underline{v}_2; M) * \dots * p(\underline{v}_L; M), \quad (2.2-27)$$

or

$$Pr\{\underline{\lambda}; M, L\} = \sum_{v_1} \sum_{v_2} \dots \sum_{v_L} Pr\{\underline{v}_1; M\} \dots Pr\{\underline{v}_L; M\} \delta(\underline{\lambda} - \sum_{k=1}^L \underline{v}_k). \quad (2.2-28)$$

Figure 2.2-1 illustrates a programming approach for calculating this equation indirectly.

J. S. LEE ASSOCIATES, INC.

TABLE 2.2-3
JAMMING EVENT PROBABILITIES FOR M=2, L=1 TO 4

<u>ℓ</u>	<u>L=1</u>	<u>L=2</u>	<u>L=3</u>	<u>L=4</u>
0,0	π_0	π_0^2	π_0^3	π_0^4
0,1	π_1	$2\pi_0\pi_1$	$3\pi_0^2\pi_1$	$4\pi_0^3\pi_1$
1,0	π_1	$2\pi_0\pi_1$	$3\pi_0^2\pi_1$	$4\pi_0^3\pi_1$
1,1	π_2	$2\pi_0\pi_2 + 2\pi_1^2$	$3\pi_0^2\pi_2 + 6\pi_0\pi_1^2$	$4\pi_0^3\pi_2 + 12\pi_0^2\pi_1^2$
0,2		π_1^2	$3\pi_0\pi_1^2$	$6\pi_0^2\pi_1^2$
1,2		$2\pi_1\pi_2$	$6\pi_0\pi_1\pi_2 + 3\pi_1^3$	$12\pi_0^2\pi_1\pi_2 + 12\pi_0\pi_1^3$
2,0		π_1^2	$3\pi_0\pi_1^2$	$6\pi_0^2\pi_1^2$
2,1		$2\pi_1\pi_2$	$6\pi_0\pi_1\pi_2 + 3\pi_1^3$	$12\pi_0^2\pi_1\pi_2 + 12\pi_0\pi_1^3$
2,2		π_2^2	$3\pi_0\pi_2^2 + 6\pi_1^2\pi_2$	$6\pi_0^2\pi_2^2 + 24\pi_0\pi_1^2\pi_2 + 6\pi_1^4$
0,3			π_1^3	$4\pi_0\pi_1^3$
1,3			$3\pi_1^2\pi_2$	$12\pi_0\pi_1^2\pi_2 + 4\pi_1^4$
2,3			$3\pi_1\pi_2^2$	$12\pi_0\pi_1\pi_2^2 + 12\pi_1^3\pi_2$
3,0			π_1^3	$4\pi_0\pi_1^3$
3,1			$3\pi_1^2\pi_2$	$12\pi_0\pi_1^2\pi_2 + 4\pi_1^4$
3,2			$3\pi_1\pi_2^2$	$12\pi_0\pi_1\pi_2^2 + 12\pi_1^3\pi_2$
3,3			π_2^3	$4\pi_0\pi_2^3 + 12\pi_1^2\pi_2^2$
0,4				π_1^4
1,4				$4\pi_1^3\pi_2$
2,4				$6\pi_1^2\pi_2^2$
3,4				$4\pi_1\pi_2^3$
4,0				π_1^4
4,1				$4\pi_1^3\pi_2$
4,2				$6\pi_1^2\pi_2^2$
4,3				$4\pi_1\pi_2^3$
4,4				π_2^4

$$Pr(\underline{\ell}) = \sum_{n=0}^{L} \binom{n, L-\frac{\ell}{2}-n, L-\ell_1-n, \ell_1+\ell_2+n-L}{\ell_1, \ell_2} \pi_0^n \pi_1^{2L-\ell_1-\ell_2-n} \pi_2^{\ell_1+\ell_2+n-L}$$

TABLE 2.2-4

PROBABILITIES OF $\underline{\lambda}$ JAMMING EVENTS FOR $M = 4$

$$\Pr \{ \underline{\lambda} ; 4, L \} = \underbrace{\sum_{n_0=0}^L \sum_{n_1=1}^L \cdots \sum_{n_{15}=0}^L}_{\text{CONSTRAINTS}} \cdot \overbrace{\sum_{n_0!n_1! \cdots n_{15}!}^L}^{\frac{L!}{n_0!n_1! \cdots n_{15}!}} \pi_0^{n_0} \pi_1^{n_1+n_2+n_4+n_8} \pi_2^{n_3+n_5+n_6+n_9+n_{10}+n_{12}} \pi_3^{n_7+n_{11}+n_{13}+n_{14}} \pi_4^{n_{15}}$$

$$\text{CONSTRAINTS : } n_1 + n_3 + n_5 + n_7 + n_9 + n_{11} + n_{13} + n_{15} = \lambda_1$$

$$n_2 + n_3 + n_6 + n_7 + n_{10} + n_{11} + n_{14} + n_{15} = \lambda_2$$

$$n_4 + n_5 + n_6 + n_7 + n_{12} + n_{13} + n_{14} + n_{15} = \lambda_3$$

$$n_8 + n_9 + n_{10} + n_{11} + n_{12} + n_{13} + n_{14} + n_{15} = \lambda_4$$

15

$$\sum_{i=0}^L n_i = L$$

J. S. LEE ASSOCIATES, INC.

Input: M, L, π_0 to π_M

Initialize all $Pr\{\underline{\varepsilon}\} = 0$

$$I_1 = 0 \text{ to } 2^M - 1$$

v_{m1} = mth bit of I_1 , $m = 1$ to M

$$w_1 = \sum_m v_{m1}$$

$$I_2 = 0 \text{ to } 2^M - 1$$

v_{m2} = mth bit of I_2 , $m = 1$ to M

$$w_2 = \sum_m v_{m2}$$

.

.

$$I_L = 0 \text{ to } 2^M - 1$$

v_{mL} = mth bit of I_L , $m = 1$ to M

$$w_L = \sum_m v_{mL}$$

$$\lambda_m = \sum_k v_{mk}, m = 1 \text{ to } M$$

$$I(\underline{\varepsilon}) = \sum_m \lambda_m L^{m-1}$$

Increment $Pr\{\underline{\varepsilon}\} = Pr\{I(\underline{\varepsilon})\}$
by $-(w_1) - (w_2) \dots - (w_L)$

} skip if any

$\lambda_m > \lambda_{m+1}$, $m = 2$ to $M-1$

} to get ordered vector output

FIGURE 2.2-1 PROGRAM STRUCTURE FOR CALCULATING JAMMING EVENT PROBABILITIES INDIRECTLY

J. S. LEE ASSOCIATES, INC.

This can also be done iteratively, using the fact that

$$\begin{aligned} \Pr\{\underline{\alpha}; M, L\} &= \sum_{\underline{\alpha}} \sum_{\underline{v}_L} \Pr\{\underline{\alpha}; M, L-1\} \Pr\{\underline{v}_L; M\} \delta(\underline{\alpha} - \underline{\alpha} - \underline{v}_L) \\ &= \sum_{\underline{v}_L} \Pr\{\underline{\alpha} - \underline{v}_L; M, L-1\} \Pr\{\underline{v}_L; M\}. \end{aligned} \quad (2.2-29)$$

To accomplish the vector additions needed for the convolution, we may encode the $M \times 1$ vectors $\underline{\alpha}$ and \underline{v}_L into a number using the form

$$I(\underline{\alpha}) = \alpha_1 + b\alpha_2 + b^2\alpha_3 + \dots + b^{M-1}\alpha_M, \quad (2.2-30a)$$

where $b \geq L$ is an integer base number, supporting the relationship

$$I(\underline{\alpha}_1 + \underline{\alpha}_2) = I(\underline{\alpha}_1) + I(\underline{\alpha}_2). \quad (2.2-30b)$$

In this manner the convolution in (2.2-29) can be done using

$$\begin{aligned} \Pr\{\underline{\alpha}; M, L\} &= \sum_{I(\underline{\alpha})} \sum_{I(\underline{v}_L)} \Pr\{I(\underline{\alpha}); M, L-1\} \Pr\{I(\underline{v}_L); M\} \\ &\quad \times \delta[I(\underline{\alpha}) - I(\underline{\alpha}) - I(\underline{v}_L)]. \\ &= \sum_{I(\underline{v}_L)} \Pr\{I(\underline{\alpha}) - I(\underline{v}_L); M, L-1\} \Pr\{I(\underline{v}_L); M\}. \end{aligned} \quad (2.2-31)$$

2.3 TOTAL PROBABILITY OF ERROR

In terms of the conditional probability of symbol error, given a jamming event defined by the vector $\underline{\alpha}$, and the probabilities of the jamming events, we now can write the total probability of error as

$$\begin{aligned} P_s(e) &= \sum_{\underline{\alpha}} \Pr\{\underline{\alpha}\} P_s(e|\underline{\alpha}) \\ &= \sum_{\underline{\alpha}} \binom{M-1}{n_0, n_1, \dots, n_L} \Pr\{\underline{\alpha}'\} P_s(e|\underline{\alpha}'). \end{aligned} \quad (2.3-1)$$

J. S. LEE ASSOCIATES, INC.

This formulation utilizes the fact that $\underline{\ell}$ vectors formed by permutations of the nonsignal channel elements $\{e_m, m>1\}$ of the ordered $\underline{\ell}$ vector are equiprobable, and that the conditional error probability is the same for each permutation.

2.4 FH/RMFSK HARD DECISION ANALYSIS

In addition to studying the performance of L hops/symbols soft decision receivers for various FH/RMFSK combining schemes, we shall calculate the performance when L M-ary hard decisions are combined to produce a final symbol decision. This configuration is in itself a form of ECCM processing, as will be shown in later sections.

2.4.1 Formulation of Error Probability.

Under an M-ary hard decision approach, shown previously as Figure 1.3-3, on the kth hop the decision variables $\{z_{mk}\}$ are compared to find the largest; the signal is assumed to be present in the channel with the largest decision variable. The per hop or "hard" symbol decision can be thought of as selecting one of M vectors $\{D_1, D_2, \dots, D_M\}$, where

$$D_m = (D_{1m}, D_{2m}, \dots, D_{Mm}) \quad (2.4-1a)$$

with

$$D_{im} = \begin{cases} 1, & i=m \\ 0, & i \neq m \end{cases} \quad m=1, 2, \dots, M. \quad (2.4-1b)$$

The hard symbol decision on the kth hop then can be expressed as the vector

$$d_k = (d_{1k}, d_{2k}, \dots, d_{Mk}) \triangleq D_{m^*} \quad (2.4-2a)$$

where m^* is chosen such that

$$z_{m^*k} = \max_m(z_{mk}). \quad (2.4-2b)$$

J. S. LEE ASSOCIATES, INC.

The components of the per-hop decision vector \underline{d}_k are accumulated over the L hops of the symbol to produce the final, discrete decision variables

$$d_m = \sum_{k=1}^L d_{mk}; m = 1, 2, \dots, M; \quad (2.4-3a)$$

or, in vector notation, the final discrete-valued decision vector

$$\underline{d} = \sum_{k=1}^L \underline{d}_k. \quad (2.4-3b)$$

The error probability can be formulated as

$$P_s(e) = 1 - P_s(\text{correct decision} \equiv C)$$

$$\begin{aligned} &= 1 - \sum_{\underline{\ell}} P_s(C|\underline{\ell}) \Pr\{\underline{\ell}\} \\ &= 1 - \sum_{\underline{\ell}} \Pr\{\underline{\ell}\} \sum_{\underline{n} \in \Omega_C} \Pr\{\underline{d} = \underline{n} | \underline{\ell}\}, \end{aligned} \quad (2.4-4)$$

where $\underline{\ell} = (\ell_1, \ell_2, \dots, \ell_M)$ describes the jamming events, and Ω_C is the set of decision vectors which produce a correct decision.

Since the components of \underline{d} are discrete-valued, there exists the possibility of a tie between the signal channel's final decision variable value and that of one or more non-signal channels. Thus the error expression (2.4-4) should be modified to

$$P_s(e) = 1 - \sum_{\underline{\ell}} \Pr\{\underline{\ell}\} \sum_{\underline{n} \in \Omega_C} h(\underline{n}) \Pr\{\underline{d} = \underline{n} | \underline{\ell}\}, \quad (2.4-5a)$$

where $h(\underline{n}) = (\#\text{channels equal to maximum})^{-1}$, (2.4-5b)

assuming that a randomized decision is made when there is a tie. For example, if three channels (including the signal channel) are equal to the maximum value, then $h(\underline{n}) = 1/3$.

J. S. LEE ASSOCIATES, INC.

2.4.2 Explicit Form of Error Probability.

Interchanging the summations in the probability of a correct symbol decision gives

$$P_s(c) = \sum_{\underline{n} \in \Omega_c} h(\underline{n}) \sum_{\underline{\ell}} \Pr\{\underline{\ell}\} \Pr\{\underline{d} = \underline{n} | \underline{\ell}\}. \quad (2.4-6)$$

Now, since the jamming event vector $\underline{\ell}$ is related to the jamming event matrix $[\underline{v}]$ by

$$\underline{\ell} = \sum_{k=1}^L \underline{v}_k, \quad (2.4-7)$$

where \underline{v}_k is the kth column of $[\underline{v}]$, the summation over $\underline{\ell}$ can be replaced by

$$\begin{aligned} & \sum_{\underline{v}_1} \sum_{\underline{v}_2} \dots \sum_{\underline{v}_L} \Pr\{[\underline{v}]\} \Pr\{\underline{d} = \underline{n} | [\underline{v}]\} \\ &= \sum_{\underline{v}_1} \Pr\{\underline{v}_1\} \sum_{\underline{v}_2} \Pr\{\underline{v}_2\} \dots \sum_{\underline{v}_L} \Pr\{\underline{v}_L\} \Pr\{\underline{d} = \underline{n} | \underline{v}_1, \underline{v}_2, \dots, \underline{v}_L\}. \end{aligned} \quad (2.4-8)$$

In this expression we use the fact that the $\{\underline{v}_k\}$ are statistically independent. It is also true that the individual hop decisions $\{d_k\}$ are independent, so we can expand (2.4-8) further to obtain

$$\begin{aligned} & \sum_{\underline{v}_1} \Pr\{\underline{v}_1\} \Pr\{d_1 | \underline{v}_1\} \sum_{\underline{v}_2} \Pr\{\underline{v}_2\} \Pr\{d_2 | \underline{v}_2\} \\ & \dots \sum_{\underline{v}_L} \Pr\{\underline{v}_L\} \Pr\{d_L | \underline{v}_L\} \delta\left(\sum_k d_k, \underline{n}\right), \end{aligned} \quad (2.4-9)$$

J. S. LEE ASSOCIATES, INC.

where $\delta(\underline{a}, \underline{b})$ is a vector version of the Kronecker delta function:

$$\delta(\underline{a}, \underline{b}) \triangleq \begin{cases} 1 & \text{if } \underline{a} = \underline{b} \\ 0 & \text{if } \underline{a} \neq \underline{b} \end{cases}. \quad (2.4-10)$$

Recognizing that the sums over the individual $\{\underline{v}_k\}$ are simply averages, we can write

$$P_s(e) = 1 - \sum_{\underline{n} \in \Omega_c} h(n) \Pr\{\underline{d}_1\} \Pr\{\underline{d}_2\} \dots \Pr\{\underline{d}_L\} \delta(\underline{d}, \underline{n}), \quad (2.4-11)$$

where $\Pr\{\underline{d}_k\}$ for the kth hop is the average of the (discrete) probability distribution for the hop decision over the jamming events for the kth hop, \underline{v}_k . Assuming without loss of generality that the first channel is the signal channel, these averages are

$$\Pr\{\underline{d}_k = D_1\} \equiv p = 1 - P_s(e; \gamma, \frac{1}{L} \cdot \frac{E_s}{N_o}, \frac{1}{L} \cdot \frac{E_s}{N_j}, L=1) \triangleq 1 - P_1 \quad (2.4-12a)$$

$$\begin{aligned} \Pr\{\underline{d}_k = D_m\} \equiv q &= (1-p)/(M-1) \\ &= P_1/(M-1), \quad m \geq 2. \end{aligned} \quad (2.4-12b)$$

Finally, using the function

$$h(n) = \begin{cases} 1 & \text{if } \underline{n} \in \Omega_c \\ 0 & \text{if } \underline{n} \in \Omega_c' \end{cases} \quad (2.4-13)$$

J. S. LEE ASSOCIATES, INC.

as a "mask" for the correct decisions, we can write (2.4-11) more explicitly as

$$\begin{aligned}
 P_s(e) &= 1 - \underbrace{\sum_{n_1=0}^L \sum_{n_2=0}^L \cdots \sum_{n_M=0}^L}_{\sum_i n_i = L} H(\underline{n}) h(\underline{n}) \left(n_1, n_2, \dots, n_M \right) p^{n_1} q^{L-n_1} \\
 &= 1 - \sum_{n_1=0}^L \sum_{n_2=0}^{n_1} \cdots \sum_{n_M=0}^{n_1} h(\underline{n}) \left(n_1, n_2, \dots, n_M \right) p^{n_1} q^{L-n_1} \delta(\sum_i n_i, L).
 \end{aligned} \tag{2.4-14}$$

2.4.2.1 Special case: $L = 2$.

For $L = 2$, (2.4-14) reduces to

$$\begin{aligned}
 P_s(e; \gamma, \frac{E_s}{N_0}, \frac{E_s}{N_J}, L=2) &= 1 - p^2 - (M-1)pq \\
 &= 1 - p^2 - p P_1 = P_1 \\
 &= P_s(e; \gamma, \frac{1}{2} \frac{E_s}{N_0}, \frac{1}{2} \frac{E_s}{N_J}, L=1);
 \end{aligned} \tag{2.4-15}$$

that is, the hard decision receiver is uniformly 3 dB worse for $L = 2$ than for $L = 1$, for any value of M .

2.4.2.2 Special case: $L = 3$.

For $L = 3$, (2.4-14) reduces to

$$P_s(e; \gamma, \frac{E_s}{N_0}, \frac{E_s}{N_J}, L=3) = \frac{1}{M-1} P_1^2 (2M - 1 - MP_1), \tag{2.4-16a}$$

with

$$P_1 = P_s(e; \gamma, \frac{1}{3} \frac{E_s}{N_0}, \frac{1}{3} \frac{E_s}{N_J}, L=1). \tag{2.4-16b}$$

J. S. LEE ASSOCIATES, INC.

2.4.2.3 Special case: $L = 4$.

For $L = 4$, (2.4-14) reduces to

$$P_s(e; \gamma, \frac{E_s}{N_0}, \frac{E_s}{N_J}, L=3) = \frac{1}{M-1} p_1^2 \left[3 + \frac{p_1}{M-1} (3M^2 - 9M + 4) - 2M \cdot \frac{M-2}{M-1} p_1^2 \right] \quad (2.4-17a)$$

with

$$p_1 = P_s(e; \gamma, \frac{1}{4} \frac{E_s}{N_0}, \frac{1}{4} \frac{E_s}{N_J}, L=1). \quad (2.4-17b)$$

Note that for $M = 2$, (2.4-16a) and (2.4-17a) both give $P_s = p_1^2(3-2p_1)$; this implies that the $L = 4$ hard decision performance is uniformly $10 \log_{10} (\frac{4}{3}) = 1.25$ dB worse than that for $L = 3$ when $M = 2$.

2.4.2.4 Special case: $M = 2$.

For $M = 2$, (2.3-14) reduces to

$$P_b(e) = \begin{cases} 1 - \sum_{n_1=1}^{\frac{L}{2}} \binom{L}{n_1} p^{n_1} q^{L-n_1} - \frac{1}{2} \binom{L}{L/2} p^{L/2} q^{L/2}, & L \text{ even} \\ 1 - \sum_{n_1=(L+1)/2}^L \binom{L}{n_1} p^{n_1} q^{L-n_1}, & L \text{ odd.} \end{cases} \quad (2.4-18)$$

J. S. LEE ASSOCIATES, INC.

3.0 FH/RMFSK PERFORMANCE USING SQUARE-LAW LINEAR COMBINING RECEIVER

In this section we consider the case of the generic receiver shown in Figure 2.2-1 when the envelope samples are processed using the function

$$f(x_{mk}) = x_{mk}^2 \equiv z_{mk}. \quad (3.0-1)$$

That is, the decision statistics $\{z_m\}$ are the unweighted linear combinations or sums of samples of the squared envelopes in each channel over multiple (L) hops. For non-hopping systems, this receiver is known to give good performance when the signal is subject to Rayleigh fading, L being the order of diversity which can be chosen to optimize performance for a given SNR.

3.1 ERROR PROBABILITY ANALYSIS

In Section 2.2 it was shown that the envelope samples $\{x_{mk}\}$ are σ_{mk} times a Rician random variable for the signal channel ($m=1$) and σ_{mk} times a Rayleigh random variable for the non-signal channels ($m>1$), where the value of σ_{mk} depends upon whether the channel is jammed or not. Therefore, for the square-law linear combining FH/RMFSK receiver, the hop decision statistics, which are the squares of the envelope samples, are σ_{mk}^2 times chi-squared random variables with two degrees of freedom. For the signal channel the noncentrality parameter is

$$\lambda_k = 2\rho_k = 2S/\sigma_{1k}^2. \quad (3.1-1)$$

The probability density function (pdf) for the signal channel samples is

$$p_{z_{1k}}(z; \rho_k, \sigma_{1k}^2) = \frac{1}{2\sigma_{1k}^2} \exp \left\{ -\rho_k - \frac{z}{2\sigma_{1k}^2} \right\} I_0(\sqrt{2\rho_k z / \sigma_{1k}^2}), \quad (3.1-2)$$

while that for the non-signal channel samples is

J. S. LEE ASSOCIATES, INC.

$$p_{z_{mk}}(\alpha; \sigma_{mk}^2) = \frac{1}{2\sigma_{mk}^2} \exp \left\{ -\frac{\alpha}{2\sigma_{mk}^2} \right\}, \quad m > 1. \quad (3.1-3)$$

3.1.1 Distribution of the Decision Statistics.

The accumulated decision statistics $\{z_m\}$ can be expressed as

$$z_1 = \sum_{k=1}^L \sigma_{1k}^2 \chi^2(2; \lambda_k) \quad (3.1-4a)$$

and

$$z_m = \sum_{k=1}^L \sigma_{mk}^2 \chi^2(2) \quad (3.1-4b)$$

where $\chi^2(n)$ denotes a chi-squared random variable with n degrees of freedom and $\chi^2(n; \lambda)$ denotes a noncentral chi-squared random variable with n degrees of freedom and a noncentrality parameter λ . It is well known that sums of equally weighted chi-squared variables yield chi-squared variables:

$$\sum_{k=1}^L \chi^2(n_k; \lambda_k) = \chi^2\left(\sum_{k=1}^L n_k; \sum_{k=1}^L \lambda_k\right). \quad (3.1-5)$$

This fact can be applied to (3.1-4) by recalling that for a given jamming event, ℓ_m out of L hops in a given channel are jammed. Thus

$$z_1 \approx \sigma_N^2 \chi^2[2(L-\ell_1); 2(L-\ell_1)S/\sigma_N^2] + \sigma_T^2 \chi^2(2\ell_1; 2\ell_1 S/\sigma_T^2) \quad (3.1-6a)$$

and

$$z_m = \sigma_N^2 \chi^2[2(L-\ell_m)] + \sigma_T^2 \chi^2(2\ell_m), \quad m > 2. \quad (3.1-6b)$$

In Appendix A it is shown that the pdf for the normalized variable

$u_1 = z_1 / \sigma_N^2$ is

J. S. LEE ASSOCIATES, INC.

$$p_{u_1}(\alpha) = \begin{cases} p_{\chi^2}(\alpha; 2L, 2L\rho_N) & \ell_1 = 0; \\ \frac{1}{K} p_{\chi^2}(\alpha/K; 2L, 2L\rho_T), & \ell_1 = L; \\ \sum_{n=0}^{\infty} c_n p_{\chi^2}[\alpha; 2L + 2n, 2(L-\ell_1)\rho_N], & 0 < \ell_1 < L; \end{cases} \quad (3.1-7)$$

using

$$\rho_N \equiv S/\sigma_N^2, \rho_T \equiv S/\sigma_T^2, K = \sigma_T^2/\sigma_N^2 \quad (3.1-7d)$$

and where

$$c_n = e^{-\ell_1 \rho_T} \left(\frac{K-1}{K}\right)^n \left(\frac{1}{K}\right)^{\ell_1} \mathcal{L}_n^{\ell_1-1} \left[-\frac{\ell_1 \rho_T}{K-1} \right], \ell_1 \geq 1. \quad (3.1-7e)$$

In (3.1-7e), the function $\mathcal{L}_n^{\ell_1}(x)$ is the generalized Laguerre polynomial.

For the non-signal channels, substitution of $\rho_N = \rho_T = 0$ in (3.1-7) yields, for $u_m = z_m/\sigma_N^2$ ($m > 1$),

$$p_{u_m}(\alpha) = \begin{cases} p_{\chi^2}(\alpha; 2L), \ell_m = 0; & (3.1-8a) \\ \frac{1}{K} p_{\chi^2}(\alpha/K; 2L), \ell_m = L; & (3.1-8b) \\ \sum_{n=0}^{\infty} b_n p_{\chi^2}(\alpha; 2L+2n), 0 < \ell_m < L; & (3.1-8c) \end{cases}$$

where

$$b_n = \left(\frac{K-1}{K}\right)^n \left(\frac{1}{K}\right)^{\ell_m} \binom{n+\ell_m-1}{n}. \quad (3.1-8d)$$

In (3.1-7) and (3.1-8) the chi-squared pdf's are, for $2n$ degrees of freedom,

$$p_{\chi^2}(\alpha; 2n, 2\rho) = \frac{1}{2} \exp\left\{-\frac{\alpha}{2} - \rho\right\} \left(\frac{\alpha}{2\rho}\right)^{(n-1)/2} I_{n-1}(\sqrt{2\rho\alpha}) \quad (3.1-9a)$$

$$= \frac{1}{2} e^{-\rho/2} (\rho/2)^{n-1} / r(n), \rho = 0 \quad (3.1-9b)$$

J. S. LEE ASSOCIATES, INC.

where $I_m(x)$ is the modified Bessel function of the first kind and order m and $\Gamma(\cdot)$ is the gamma function.

3.1.2 Formulation of the Conditional Error Probability.

From Section 2.2, the conditional symbol error probability is

$$\begin{aligned}
 P_s(e|v) &= P_s(e|\ell_1, \ell_2, \dots, \ell_M) \\
 &= 1 - \Pr\{z_2 < z_1, z_3 < z_1, \dots, z_M < z_1 | \underline{\ell}\} \\
 &= 1 - \Pr\{u_2 < u_1, u_3 < u_1, \dots, u_M < u_1 | \underline{\ell}\} \\
 &= 1 - \int_0^\infty d\alpha p_{u_1}(\alpha) \prod_{m=2}^M \int_0^\alpha d\beta_m p_{u_m}(\beta_m). \quad (3.1-10)
 \end{aligned}$$

From (3.1-8) and (3.1-9),

$$F_L(\alpha; \ell_m) \triangleq \int_0^\alpha d\beta_m p_{u_m}(\beta_m) = \begin{cases} 1 - \Gamma(L; \alpha/2)/\Gamma(L), & \ell_m = 0; \\ 1 - \Gamma(L; \alpha/2K)/\Gamma(L), & \ell_m = L; \\ 1 - \sum_{n=0}^{\infty} b_n \Gamma(L+n; \alpha/2)/\Gamma(L+n), & 0 < \ell_m < L; \end{cases} \quad (3.1-11a)$$

$$F_L(\alpha; \ell_m) \triangleq \int_0^\alpha d\beta_m p_{u_m}(\beta_m) = \begin{cases} 1 - \Gamma(L; \alpha/2K)/\Gamma(L), & \ell_m = L; \\ 1 - \sum_{n=0}^{\infty} b_n \Gamma(L+n; \alpha/2)/\Gamma(L+n), & 0 < \ell_m < L; \end{cases} \quad (3.1-11b)$$

$$F_L(\alpha; \ell_m) \triangleq \int_0^\alpha d\beta_m p_{u_m}(\beta_m) = \begin{cases} 1 - \sum_{n=0}^{\infty} b_n \Gamma(L+n; \alpha/2)/\Gamma(L+n), & 0 < \ell_m < L; \end{cases} \quad (3.1-11c)$$

where $\Gamma(n; t)$ is the incomplete gamma function,

$$\Gamma(n; t) = \int_t^\infty dx e^{-x} x^{n-1} = r(n) e^{-t} \sum_{r=0}^{n-1} \frac{t^r}{r!}. \quad (3.1-12)$$

Formally, (3.1-10) can be written

$$P_s(e|\underline{\ell}) = 1 - \int_0^\infty d\alpha p_{u_1}(\alpha) [F_L(\alpha; 0)]^{n_0} [F_L(\alpha; 1)]^{n_1} \dots [F_L(\alpha; L)]^{n_L}, \quad (3.1-13a)$$

where n_i is the number of non-signal channels with $\ell_m = i$, and it is true that

$$n_0 + n_1 + \dots + n_L = M - 1. \quad (3.1-13b)$$

J. S. LEE ASSOCIATES, INC.

3.1.3 Powers of Non-Signal Channel Probabilities.

We now show that the probabilities $F_L(\alpha; \xi_m)$ in the general expression (3.1-13a) can be written in terms of power series.

For $\xi_m = 0$, from (3.1-11) and (3.1-12)

$$\begin{aligned}
 [F_L(\alpha; 0)]^{n_0} &= \left[1 - e^{-\alpha/2} \sum_{r=0}^{L-1} \frac{(\alpha/2)^r}{r!} \right]^{n_0} \\
 &= \sum_{r_0=0}^{n_0} \binom{n_0}{r_0} (-1)^{r_0} e^{-r_0 \alpha/2} \left[\sum_{r=0}^{L-1} \frac{(\alpha/2)^r}{r!} \right]^{r_0} \\
 &= \sum_{r_0=0}^{n_0} \binom{n_0}{r_0} (-1)^{r_0} e^{-r_0 \alpha/2} \sum_{k_0=0}^{r_0(L-1)} \frac{C(k_0, r_0)}{k_0!} \left(\frac{\alpha}{2}\right)^{k_0}, \quad (3.1-14)
 \end{aligned}$$

where the coefficients $C(k_0, r_0)$ are [1, Appendix 4A] the functions

$$C(0, r) = 1 \quad (3.1-15a)$$

$$\begin{aligned}
 C(k, r) &= \frac{1}{k} \sum_{n=1}^{\min(k, L-1)} \binom{k}{n} [(r+1)n - k] C(k-n, r), \\
 &\quad k > 0, \quad L \geq 2. \quad (3.1-15b)
 \end{aligned}$$

For example, when $L = 2$ the coefficients are simply

$$\begin{aligned}
 C(k, r) &= (r+1-k) C(k-1, r) \\
 &= r! / (r-k)! \quad (3.1-16a)
 \end{aligned}$$

so that the series raised to the r_0 power is

J. S. LEE ASSOCIATES, INC.

$$\sum_{k_0=0}^{r_0} \frac{r_0!}{k_0!(r_0-k_0)!} \left(\frac{\alpha}{2}\right)^{k_0} = (1+\alpha/2)^{r_0} \quad (3.1-16b)$$

as required.

Similarly, for $\ell_m = L$, we find that

$$[F_L(\alpha; L)]^{n_L} = \sum_{r_L=0}^{n_L} \binom{n_L}{r_L} (-1)^{r_L} e^{-r_L \alpha / 2K} \sum_{k_L=0}^{r_L(L-1)} \frac{C(k_L, r_L)}{k_L!} \left(\frac{\alpha}{2K}\right)^{k_L}, \quad (3.1-17)$$

using the same $C(k, r)$ function as for $\ell_m = 0$.

For $0 < \ell_M < L$, $L \geq 2$, the evaluation of the probability is more challenging, but does indeed reduce to a closed form. The development begins by recognizing that

$$\begin{aligned} F_L(\alpha; \ell_m) &= 1 - e^{-\alpha/2} \sum_{n=0}^{\infty} b_n \sum_{r=0}^{n+L-1} \frac{(\alpha/2)^r}{r!} \\ &= 1 - e^{-\alpha/2} \sum_{r=0}^{L-1} \frac{(\alpha/2)^r}{r!} - e^{-\alpha/2} \sum_{n=0}^{\infty} b_{n+1} \sum_{r=0}^n \frac{(\alpha/2)^{r+L}}{(r+L)!} \end{aligned} \quad (3.1-18)$$

where b_n is defined by (3.1-8d). The double summation in the last term can be manipulated in the following way:

$$\begin{aligned} \sum_{n=0}^{\infty} b_{n+1} \sum_{r=0}^n \frac{(\alpha/2)^{r+L}}{(r+L)!} &= \sum_{n=0}^{\infty} \sum_{r=0}^{\infty} b_{n+r+1} \frac{(\alpha/2)^{r+L}}{(r+L)!} \\ &= K^{-\ell} \sum_{r=0}^{\infty} \left(\frac{K-1}{K}\right)^{r+1} \frac{(\alpha/2)^{r+L}}{(r+L)!} {}_{\ell-1}F_1(r+\ell+1, 1; r+2; \frac{K-1}{K}). \end{aligned} \quad (3.1-19)$$

The hypergeometric function can be transformed using [2, Eq. 15.3.5] to obtain

J. S. LEE ASSOCIATES, INC.

on the summation were zero, the summation would be a Taylor's series:

$$\begin{aligned} & \sum_{n=0}^{\infty} \frac{a^n}{n!} \frac{n!}{(n+L-\ell)!} {}_1F_1(n+1; n+L-\ell+1; b) \\ &= \frac{1}{(L-\ell)!} \sum_{n=0}^{\infty} \frac{a^n}{n!} \left[\frac{d^n}{dx^n} {}_1F_1(1; L-\ell+1; x) \right] \Big|_{x=b} \end{aligned} \quad (3.1-23a)$$

$$= \left[e^{a+b} - \sum_{n=0}^{L-\ell-1} \frac{(a+b)^n}{n!} \right] \frac{1}{(a+b)^{L-\ell}} \quad (3.1-23b)$$

therefore, (3.1-22) is seen to be

$$e^{\alpha/2} - \sum_{n=0}^{L-\ell-1} \frac{(\alpha/2)^n}{n!} - \sum_{n=0}^{\ell-1} \left(\frac{1}{K} \right)^n \frac{(\alpha/2)^{n+L-\ell}}{n!} \frac{d^n}{dx^n} x^{\ell-L} \left[e^x - \sum_{r=0}^{L-\ell-1} \frac{x^r}{r!} \right] \Big|_{x=(K-1)\alpha/2K} . \quad (3.1-24)$$

Now, substituting (3.1-24) into (3.1-21) and the resulting expression into (3.1-18) gives

$$\begin{aligned} F_L(z; \ell) &= 1 - e^{-\alpha/2} \left[\sum_{r=0}^{L-1} \frac{(\alpha/2)^r}{r!} + \sum_{r=0}^{\infty} \frac{(\alpha/2)^{r+L}}{(r+L)!} \right] \\ &\quad + e^{-\alpha/2} [(3.1-24)] \\ &= 0 + e^{-\alpha/2} [(3.1-24)] \\ &= 1 - e^{-\alpha/2} \left\{ \sum_{n=0}^{L-\ell-1} \frac{(\alpha/2)^n}{n!} \right. \\ &\quad \left. + \left(\frac{\alpha}{2} \right)^{L-\ell} \sum_{n=0}^{\ell-1} \frac{(\alpha/2K)^n}{n!} \frac{d^n}{dx^n} x^{\ell-L} \left[e^x - \sum_{r=0}^{L-\ell-1} \frac{x^r}{r!} \right] \Big|_{x=(K-1)\alpha/2K} \right\} \end{aligned}$$

$$= 1 - e^{-\alpha/2} \left\{ \sum_{n=0}^{L-\ell-1} \frac{(\alpha/2)^n}{n!} \left[1 - \left(\frac{K}{K-1} \right)^{L-\ell-n} \sum_{r=0}^{\ell-1} \binom{L-\ell+r-n-1}{r} \frac{(-1)^r}{(K-1)^r} \right] \right\}$$

$$- e^{-\alpha/2K} \left(\frac{K}{K-1} \right)^{L-\ell} \sum_{n=0}^{\ell-1} \frac{(\alpha/2K)^n}{n!} \sum_{r=0}^{\ell-1-n} \binom{L-\ell+r-1}{r} \left(\frac{-1}{K-1} \right)^r. \quad (3.1-25)$$

For example, if $L = 2$ and $\ell = 1$,

$$\begin{aligned} F_2(\alpha; 1) &= 1 - e^{-\alpha/2} \left\{ 1 - \frac{K}{K-1} \right\} - e^{-\alpha/2K} \left(\frac{K}{K-1} \right) \\ &= 1 - \frac{1}{K-1} \left\{ K e^{-\alpha/2K} - e^{-\alpha/2} \right\}. \end{aligned} \quad (3.1-26)$$

By direct algebraic calculation of (3.1-25), it can be shown that $F_L(\alpha, \ell)$ is of the form

$$F_L(\alpha; \ell) = 1 - \frac{1}{(K-1)^{L-1}} \left\{ e^{-\alpha/2K} f_1(\alpha; \ell, L) + e^{-\alpha/2} f_2(\alpha; \ell, L) \right\}, \quad (3.1-27)$$

where $f_1(\alpha; \ell, L)$ and $f_2(\alpha; \ell, L)$ are the polynomials given in Table 3.1-1.

Therefore

$$\begin{aligned} [F_L(\alpha; \ell)]^{n_\ell} &= \sum_{r_\ell=0}^{n_\ell} \binom{n_\ell}{r_\ell} \left[\frac{-1}{(K-1)^{L-1}} \right]^{r_\ell} \\ &\cdot \left[e^{-\alpha/2K} f_1(\alpha; \ell, L) + e^{-\alpha/2} f_2(\alpha; \ell, L) \right]^{r_\ell} \end{aligned}$$

TABLE 3.1-1 POLYNOMIALS FOR $F_L(\alpha; \ell)$

L	ℓ	$f_1(\alpha; \ell, L)$	$f_2(\alpha; \ell, L)$
1	0	0	1
	1	1	0
2	0	0	$(K-1)(1+\alpha/2)$
	1	K	-1
	2	$(K-1)(1+\alpha/2K)$	0
3	0	0	$(K-1)^2(1+\alpha/2+\alpha^2/8)$
	1	K^2	$-(2K-1+(K-1)\alpha/2)$
	2	$K(K-2)+(K-1)\alpha/2$	1
	3	$(K-1)^2[1+\alpha/2K+\alpha^2/8K^2]$	0
4	0	0	$(K-1)^3(1+\alpha/2+\alpha^2/8+\alpha^3/48)$
	1	K^3	$-(3K^2-3K+1+(2K^2-3K+1)\alpha/2+(K-1)^2\alpha^2/8)$
	2	$K^3-3K^2+K(K-1)\alpha/2$	$3K-1+(K-1)\alpha/2$
	3	K^3-3K^2+3K $+(K^2-3K+2)\alpha/2$ $+(K-1)^2\alpha^2/8K$	-1
	4	$(K-1)^3(1+\alpha/2K+\alpha^2/8K^2$ $+\alpha^3/48K^3)$	0

$$= \sum_{r_\ell=0}^{n_\ell} \binom{n_\ell}{r_\ell} \left[\frac{-1}{(K-1)^{L-1}} \right]^{r_\ell} \sum_{k_\ell=0}^{r_\ell} \binom{r_\ell}{k_\ell} e^{-(r_\ell - k_\ell)\alpha/2K - k_\ell\alpha/2} \\ \cdot [f_1(\alpha; \ell, L)]^{r_\ell - k_\ell} [f_2(\alpha; \ell, L)]^{k_\ell}, \quad 0 < \ell < L, \quad (3.1-28)$$

and the powers of the polynomials can be expressed as a higher order polynomial:

$$[f_1]^{r_\ell - k_\ell} [f_2]^{k_\ell} = \sum_{p_\ell=0}^P d(p_\ell) (\alpha/2)^{p_\ell}. \quad (3.1-29)$$

The coefficients $d(p)$ are given in Table 3.1-2 for L up to 4.

3.1.4 Expectation Over Signal Channel PDF.

Substitution of the powers of the non-signal channel probabilities into the conditional probability of symbol error equation yields the lengthy expression given in Table 3.1-3. The remaining analysis requires obtaining the expectation

$$\int_0^\infty d\alpha p_{u_1}(\alpha) e^{-a_0\alpha/2} (\alpha/2)^{b_0} = E_u \left\{ e^{-au_1/2} \left(\frac{u_1}{2} \right)^{b_0} \right\}, \quad (3.1-30)$$

where a_0 and b_0 are given in Table 3.1-3. From (3.1-7a), when the signal channel is not jammed ($\ell_1 = 0$), the pdf $p_{u_1}(\alpha)$ is a straightforward noncentral chi-squared pdf, and the expectation is

$$\int_0^\infty d\alpha p_{X^2}(\alpha; 2L, 2L\rho_N) e^{-a_0\alpha/2} (\alpha/2)^{b_0}$$

TABLE 3.1-2 COEFFICIENTS FOR EQUATION (3.1-29)

L	ℓ	$P = \max p$	$d(p)$
2	1	0	$K^{r_1-k_1} (-1)^{k_1}$
3	1	k_1	$(K^2)^{r_1-k_1} (-1)^{k_1} \binom{k_1}{p} (2K-1)^{k_1-p} (K-1)^p$
3	2	r_2-k_2	$\binom{r_2-k_2}{p} [K(K-2)]^{r_2-k_2-p} (K-1)^p$
4	1	$2k_1$	$(K^3)^{r_1-k_1} (-1)^{k_1} (3K^2-3K+1)^{k_1} \cdot g(p)$ <p style="text-align: center;">where $g(0) = 1, g(1) = k_1 \cdot \frac{2K^2-3K+1}{3K^2-3K+1}$</p> $g(n) = \frac{1}{n} \left\{ (k_1+1-n) \cdot \frac{2K^2-3K+1}{3K^2-3K+1} \cdot g(n-1) \right.$ $\left. + [2(k_1+1)-n] \cdot \frac{(K-1)^2/2}{3K^2-3K+1} \cdot g(n-2) \right\}, n \geq 2$
4	2	r_2	$K^{2r_2-2k_2-p} (K-1)^p \sum_{q=\max(0, p-r_2+k_2)}^{\min(p, k_2)} \binom{r_2-k_2}{p-q} \binom{k_2}{q}$ $\cdot (K-3)^{r_2-k_2-p+q} (3K-1)^{k_2-q} K^q$
4	3	$2(r_3-k_3)$	$(-1)^{k_3} (K^3-3K^2+3K)^{r_3-k_3} g(p)$ <p style="text-align: center;">where $g(0) = 1, g(1) = (r_3-k_3) \cdot \frac{K^2-3K+2}{K^3-3K^2+3K}$</p> $g(n) = \frac{1}{n} \left\{ (r_3-k_3+1-n) \cdot \frac{K^2-3K+2}{K^3-3K^2+3K} \cdot g(n-1) \right.$ $\left. + [2(r_3-k_3+1)-n] \cdot \frac{(K-1)^2/2K}{K^3-3K^2+3K} \cdot g(n-2) \right\}, n \geq 2$

TABLE 3.1-3 EXPRESSION FOR CONDITIONAL ERROR PROBABILITY

$$\begin{aligned}
 P_S(e|\underline{\alpha}) &= 1 - \int_0^\infty d\alpha p_{\mathbf{u}_1}(\alpha) [F_L(\alpha; 0)]^{n_0} [F_L(\alpha; 1)]^{n_1} \cdots [F_L(\alpha; L-1)]^{n_{L-1}} F_L(\alpha; L)]^{n_L} \\
 &= 1 - \sum_{r_0=0}^{n_0} \binom{n_0}{r_0} (-1)^{r_0} \sum_{k_0=0}^{r_0(L-1)} \frac{c(k_0, r_0)}{k_0!} \\
 &\quad \times \sum_{r_1=0}^{n_1} \binom{n_1}{r_1} \left[\frac{-1}{(k-1)^{L-1}} \right]^{r_1} \sum_{k_1=0}^{r_1} \binom{r_1}{k_1} \\
 &\quad \times \sum_{r_{L-1}=0}^{n_{L-1}} \binom{n_{L-1}}{r_{L-1}} \left[\frac{-1}{(k-1)^{L-1}} \right]^{r_{L-1}} \sum_{k_{L-1}=0}^{r_{L-1}} \binom{r_{L-1}}{k_{L-1}} \\
 &\quad \times \int_0^\infty d\alpha p_{\mathbf{u}_1}(\alpha) e^{-\bar{a}_0 \alpha / 2} (\alpha / 2)^{b_0} \\
 &\quad \bar{a}_0 = r_0 + \frac{r_L}{K} + \sum_{\ell=1}^{L-1} \left[k_\ell + \frac{r_\ell - k_\ell}{K} \right] \\
 &\quad b_0 = k_0 + k_L + \sum_{\ell=1}^{L-1} p_\ell
 \end{aligned}$$

J. S. LEE ASSOCIATES, INC.

$$\begin{aligned}
 &= \int_0^\infty d\alpha e^{-L\rho_N - (1+a_0)\alpha/2} \left(\frac{\alpha}{2L\rho_N}\right)^{(L-1)/2} I_{L-1}(\sqrt{2L\rho_N \alpha}) \left(\frac{\alpha}{2}\right)^{b_0} \\
 &= \exp\left(\frac{-L\rho_N a_0}{1+a_0}\right) (1+a_0)^{-b_0-L} \\
 &\quad \cdot \int_0^\infty dx e^{-L\rho_N/(1+a_0)-x/2} \left[\frac{x}{2L\rho_N/(1+a_0)}\right]^{(L-1)/2} I_{L-1}\left(\sqrt{\frac{2L\rho_N x}{1+a_0}}\right) \left(\frac{x}{2}\right)^{b_0} \\
 &= \exp\left(\frac{-L\rho_N a_0}{1+a_0}\right) (1+a_0)^{-b_0-L} \cdot E\left[\left(\frac{x}{2}\right)^{b_0}\right], \tag{3.1-31}
 \end{aligned}$$

where x is distributed as a noncentral chi-squared random variable with $2L$ degrees of freedom and noncentral parameter $2L\rho_N/(1+a_0)$. The moments needed are

$$E\left[\left(\frac{x}{2}\right)^{b_0}\right] = b_0! \mathcal{L}_{b_0}^{L-1}\left(-\frac{L\rho_N}{1+a_0}\right) \tag{3.1-32a}$$

$$= \sum_{r=0}^{b_0} \binom{b_0+L-1}{b_0-r} \left(\frac{L\rho_N}{1+a_0}\right)^r \frac{1}{r!} b_0! \tag{3.1-32b}$$

where $\mathcal{L}_n^a(x)$ is the generalized Laguerre polynomial. Thus for the case of

J. S. LEE ASSOCIATES, INC.

$\ell_1 = 0$, the integral in (3.1-30) becomes the quantity

$$\exp\left(-\frac{L\rho_N a_0}{1+a_0}\right) \frac{b_0!}{(1+a_0)^{b_0+L}} \mathcal{L}_{b_0}^{L-1}\left(-\frac{L\rho_N}{1+a_0}\right). \quad (3.1-33)$$

In a similar way, when all the hops in the signal channel are jammed ($\ell_1 = L$), the integral in (3.1-30) becomes

$$\exp\left(-\frac{L\rho_T K a_0}{1+K a_0}\right) \frac{K^{b_0} b_0!}{(1+K a_0)^{b_0+L}} \mathcal{L}_{b_0}^{L-1}\left(-\frac{L\rho_T}{1+K a_0}\right). \quad (3.1-34)$$

Now, when the signal channel is jammed, but not on every hop ($0 < \ell_1 < L$), the channel pdf is a series of weighted noncentral chi-squared pdf's, as shown previously in (3.1-7c). This expectation (3.1-30) yields

$$\exp\left[-\frac{(L-\ell_1)\rho_N a_0}{1+a_0}\right] \sum_{n=0}^{\infty} c_n \frac{b_0!}{(1+a_0)^{b_0+L+n}} \mathcal{L}_{b_0}^{L+n-1}\left[-\frac{(L-\ell_1)\rho_N}{1+a_0}\right], \quad (3.1-35)$$

where the weights $\{c_n\}$ are

$$c_n = e^{-\ell_1 \rho_T} \left(\frac{K-1}{K}\right)^n K^{-\ell_1} \mathcal{L}_n^{\ell_1-1}\left[-\frac{\ell_1 \rho_T}{K-1}\right]. \quad (3.1-36)$$

In order to reduce (3.1-35) to a finite summation, it is necessary to seek an identity for

$$\sum_{n=0}^{\infty} A^n \mathcal{L}_n^a(x) \mathcal{L}_r^{n+k}(y). \quad (3.1-37)$$

J. S. LEE ASSOCIATES, INC.

To accomplish this objective, we note that [3, eq. 8.970.1]

$$\mathcal{L}_r^{n+k}(y) \triangleq \frac{1}{r!} e^y y^{-n-k} \frac{d^r}{dy^r} (e^{-y} y^{r+n+k}); \quad (3.1-38)$$

substituting in (3.1-37) yields the development

$$\begin{aligned} & \frac{1}{r!} e^y y^{-k} \sum_{n=0}^{\infty} \left(\frac{A}{y} \right)^n \mathcal{L}_n^a(x) \frac{d^r}{dy^r} (e^{-y} y^{r+n+k}) \\ &= \frac{e^y y^{-k}}{r!} \frac{\partial^r}{\partial z^r} \left[e^{-z} z^{r+k} \sum_{n=0}^{\infty} \left(\frac{Az}{y} \right)^n \mathcal{L}_n^a(x) \right] \Big|_{z=y} \\ &= \frac{e^y y^{-k}}{r!} \sum_{q=0}^r \binom{r}{q} \frac{\partial^{r-q}}{\partial z^{r-q}} (e^{-z} z^{r+k}) \frac{\partial^q}{\partial z^q} \left[\sum_{n=0}^{\infty} \left(\frac{Az}{y} \right)^n \mathcal{L}_n^a(x) \right] \Big|_{z=y} \\ &= e^y y^{-k} \sum_{q=0}^r \frac{1}{q!} e^{-z} z^{k+q} \mathcal{L}_{r-q}^{k+q}(z) \sum_{n=q}^{\infty} \left(\frac{A}{y} \right)^n \frac{n!}{(n-q)!} z^{n-q} \mathcal{L}_n^a(x) \Big|_{z=y}. \end{aligned} \quad (3.1-39)$$

The second summation, when manipulated, gives the result

$$\begin{aligned} & \left(\frac{A}{y} \right)^q \sum_{n=0}^{\infty} \frac{(n+q)!}{n!} \left(\frac{Az}{y} \right)^n \mathcal{L}_{n+q}^a(x) \\ &= \left(\frac{A}{y} \right)^q \sum_{n=0}^{\infty} \left(\frac{Az}{y} \right)^n \frac{(n+q+a)!}{n! a!} {}_1F_1(-n-q; a+1; x) \end{aligned}$$

J. S. LEE ASSOCIATES, INC.

$$= \left(\frac{A}{y}\right)^q e^x \sum_{n=0}^{\infty} \left(\frac{Az}{y}\right)^n \frac{(n+q+a)!}{n! a!} {}_1F_1(n+q+a+1; a+1; -x)$$

$$= \left(\frac{A}{y}\right)^q e^x \sum_{n=0}^{\infty} \left(\frac{Az}{y}\right)^n \frac{1}{n!} \sum_{m=0}^{\infty} \frac{(-x)^m}{m!} \frac{(n+q+a+m)!}{(a+m)!}$$

$$= \left(\frac{A}{y}\right)^q e^x \sum_{m=0}^{\infty} \frac{(-x)^m}{m!} \frac{(q+a+m)!}{(a+m)!} (1 - Az/y)^{-q-a-m-1}$$

$$= \left(\frac{A}{z}\right)^q e^x \frac{(q+a)!}{a!} (1 - Az/y)^{-q-a-1} {}_1F_1(q+a+1; a+1; \frac{-x}{1 - Az/y})$$

$$= \left(\frac{A}{y}\right)^q \frac{\exp\left(\frac{-Axz}{y-Az}\right)}{(1-Az/y)^{q+a+1}} q! \mathcal{L}_q^a\left(\frac{xy}{y-Az}\right) . \quad (3.1-40)$$

After substitution in (3.1-39), we find that

$$\sum_{n=0}^{\infty} A^n \mathcal{L}_n^a(x) \mathcal{L}_r^{n+k}(y)$$

$$= \frac{1}{(1-A)^{a+1}} \exp\left(\frac{-Ax}{1-A}\right) \sum_{q=0}^r \left(\frac{A}{1-A}\right)^q \mathcal{L}_{r-q}^{k+q}(y) \mathcal{L}_q^a\left(\frac{x}{1-A}\right). \quad (3.1-41)$$

J. S. LEE ASSOCIATES, INC.

And, substituting appropriately, the desired expectation (3.1-35) becomes

$$\exp \left[-\frac{(L-\ell_1)\rho_N a_0}{1+a_0} - \frac{\ell_1 \rho_T K a_0}{1+K a_0} \right] \frac{b_0!}{(1+a_0)^{b_0+L-\ell_1} (1+K a_0)^{\ell_1}} \\ \cdot \sum_{q=0}^{b_0} \left(\frac{K-1}{1+K a_0} \right)^q \mathcal{L}_{b_0-q}^{q+L-1} \left[-\frac{(L-\ell_1)\rho_N}{1+a_0} \right] \mathcal{L}_q^{\ell_1-1} \left[-\frac{\ell_1 \rho_T}{K-1} \cdot \frac{K(1+a_0)}{1+K a_0} \right]. \quad (3.1-42)$$

3.1.5 Special case: L=1 and M=2.

For L=1 and M=2 the FH/RMFSK total error probability is

$$P(e) = \pi_0 \cdot \frac{1}{2} e^{-\rho_N/2} + \pi_1 \cdot e^{-\rho_N/(K+1)} + \pi_2 \cdot \frac{1}{2} e^{-\rho_T/2} \quad (3.1-43a)$$

where $\rho_N = E_b/N_0$, $\rho_T = E_b/N_T$; (3.1-43b)

and $K = \sigma_T^2/\sigma_N^2 = \rho_N/\rho_T$. (3.1-43c)

J. S. LEE ASSOCIATES, INC.

3.2 NUMERICAL RESULTS

3.2.1 Soft-Decision Receiver.

Numerical results were obtained using two computational methods. In regions where the series converge rapidly enough, the form given in Table 3.1-3 and equation (3.1-42) was used for the computations. However, the presence of the term $(K-1)^L$ in the denominator of several terms causes difficulty when K is nearly equal to 1. In these cases, and on other occasions when round-off errors became excessive, the computations were performed by direct numerical integration of (3.1-10) using the densities (3.1-7) and (3.1-8) and the identity

$$1 - \int_0^{\infty} d\alpha p(\alpha) g(\alpha) = \int_0^{\infty} d\alpha p(\alpha) [1-g(\alpha)] \quad (3.2-1)$$

which holds for all density functions $p(\alpha)$ for which $p(\alpha) = 0$ if $\alpha < 0$, and hence by the properties of a p.d.f.

$$\int_0^{\infty} d\alpha p(\alpha) = 1. \quad (3.2-2)$$

Then (3.2-1) follows immediately from the fact that integration is a linear operation. The form on the right-hand side of (3.2-1) has the computational advantage that only the integrand need be computed to high accuracy, rather than the integral. For example, if $P_s(e) \approx 10^{-5}$ and we desire 4 significant digits in the answer, than the integral on the left-hand side must be computed numerically to 8-digit accuracy (e.g. 0.9999xxxx) in order to leave 4 non-zero

J. S. LEE ASSOCIATES, INC.

correct digits after subtracting from 1. But if we use the form on the right-hand side of (3.2-1) we could demand only 4-place relative accuracy from the numerical integration; the burden of many-place accuracy is placed only on the function $g(\cdot)$, which is usually much simpler and faster to compute than the overall integral. A listing of the computer program is given in Appendix D.

In Figures 3.2-1 through 3.2-4 we show the bit error probability as a function of bit energy-to-jamming density ratio with jamming fraction γ as a parameter for the case of $M=2$ (binary FSK) and $L=1,2,3$, and 4 hops per symbol (bit), respectively. In these figures the ratio of the bit energy to thermal noise density is set at 13.3525 dB, which corresponds to a bit error probability of 10^{-5} for one hop per bit in the absence of jamming. We note that for a given E_b/N_j ratio there is an optimum value of γ which maximizes the jammer's effectiveness. Furthermore, an incorrect choice of γ by the jammer can reduce the effectiveness (as measured by the communicator's bit error probability) by possibly as much as two orders of magnitude.

Figure 3.2-5 shows the envelopes of the curves in Figures 3.2-1 through 3.2-4, which represent the performance of the square-law combining receiver in worst-case partial-band noise jamming. We note that increasing L , the number of hops per bit, consistently degrades the performance. This implies that the noncoherent combining loss dominates over any diversity gain effects.

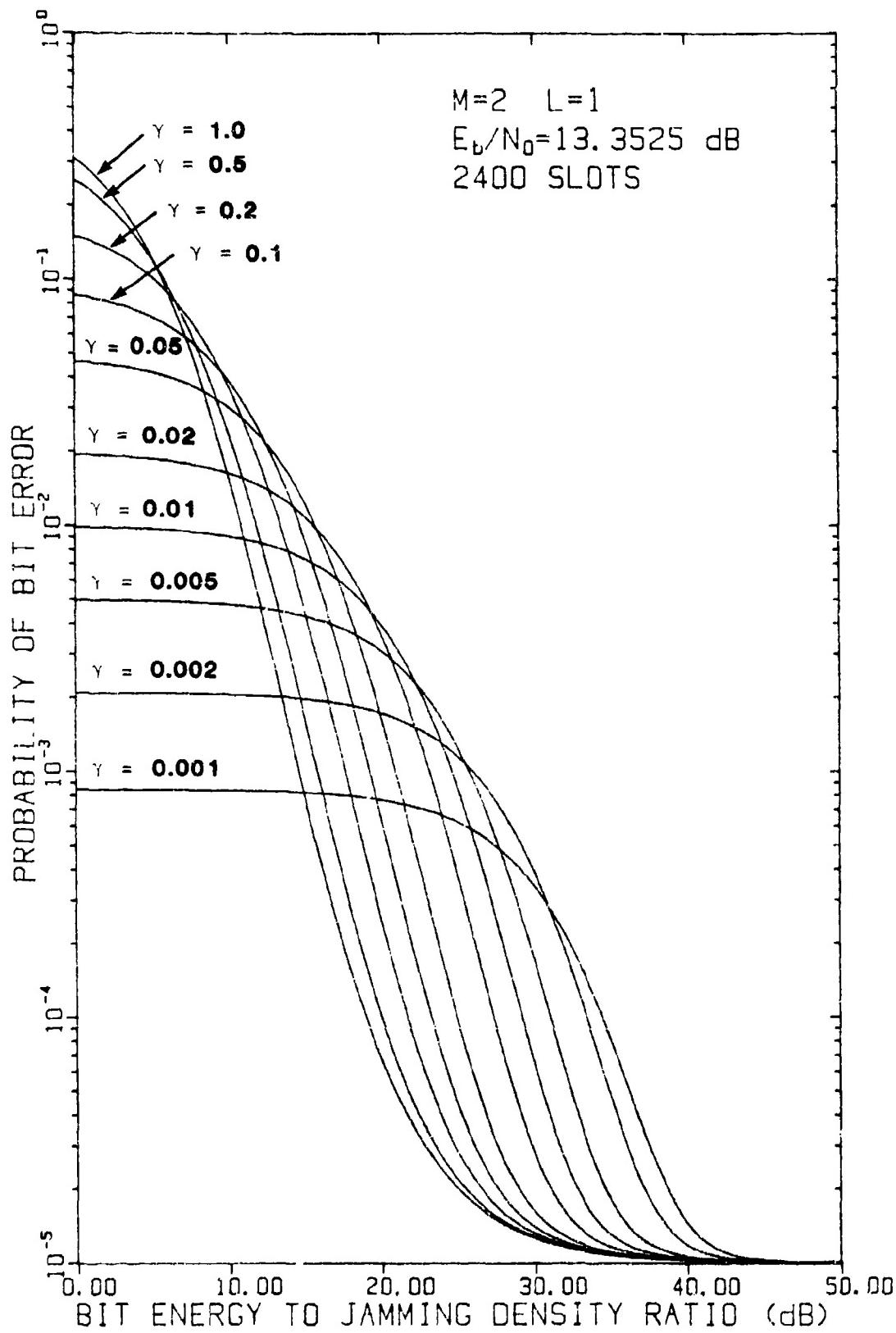


FIGURE 3.2-1 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH $M = 2$ AND $L = 1$ HOP/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.3525 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L = 1$) WITH JAMMING FRACTION γ AS A PARAMETER

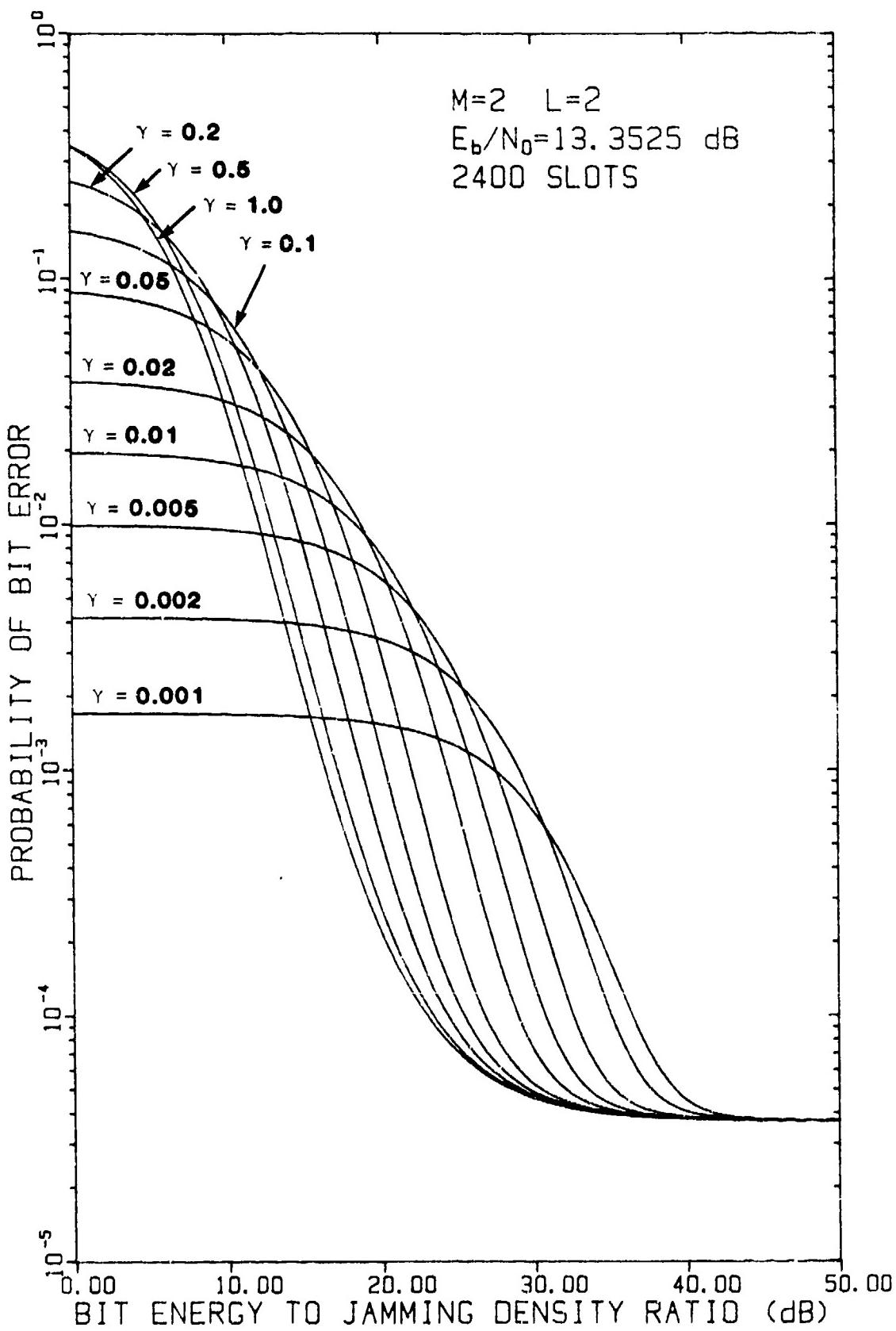


FIGURE 3.2-2 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH $M=2$ AND $L=2$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.3525 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L=1$) WITH JAMMING FRACTION γ AS A PARAMETER

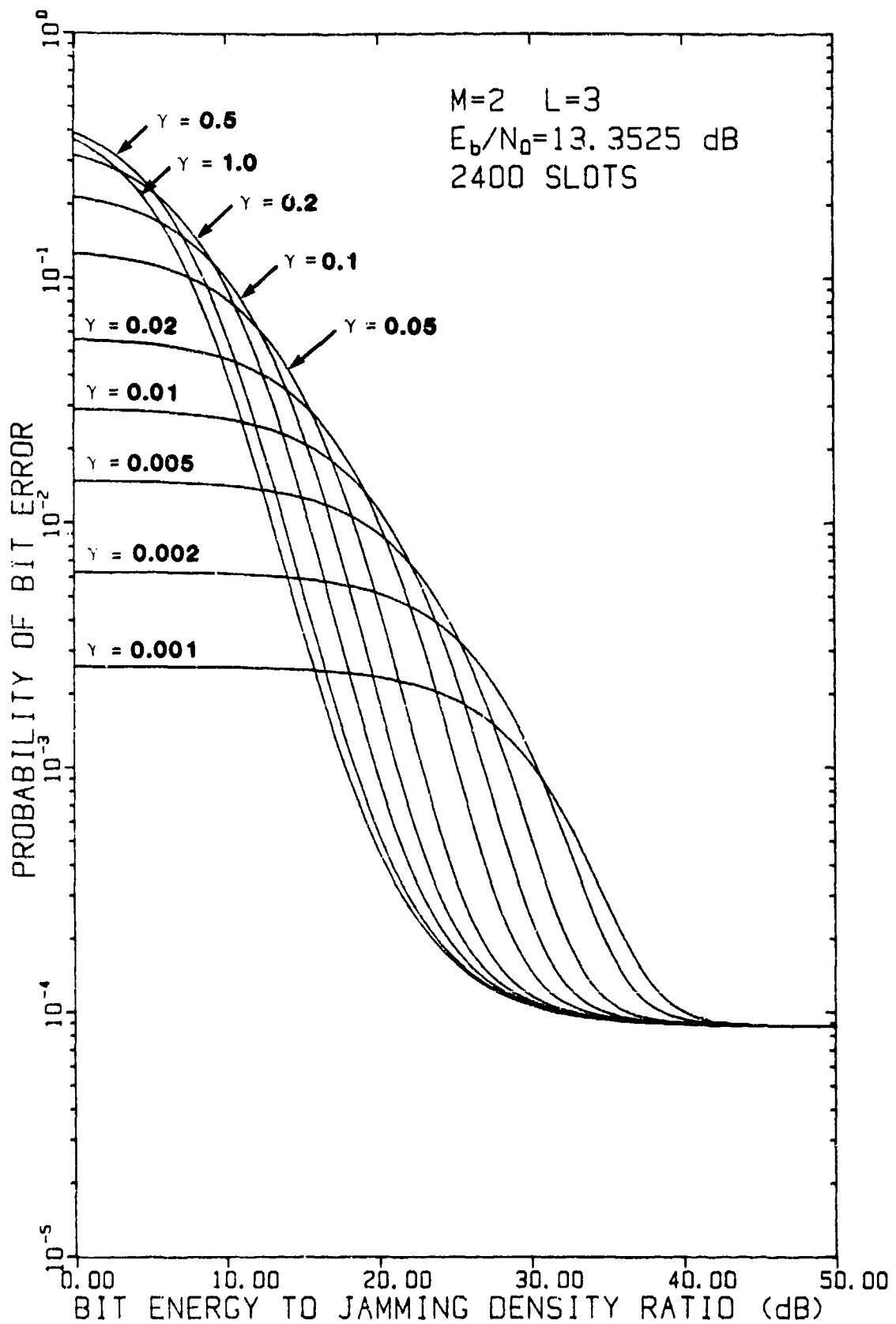


FIGURE 3.2-3 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH $M=2$ AND $L=3$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.3525 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L=1$) WITH JAMMING FRACTION γ AS A PARAMETER

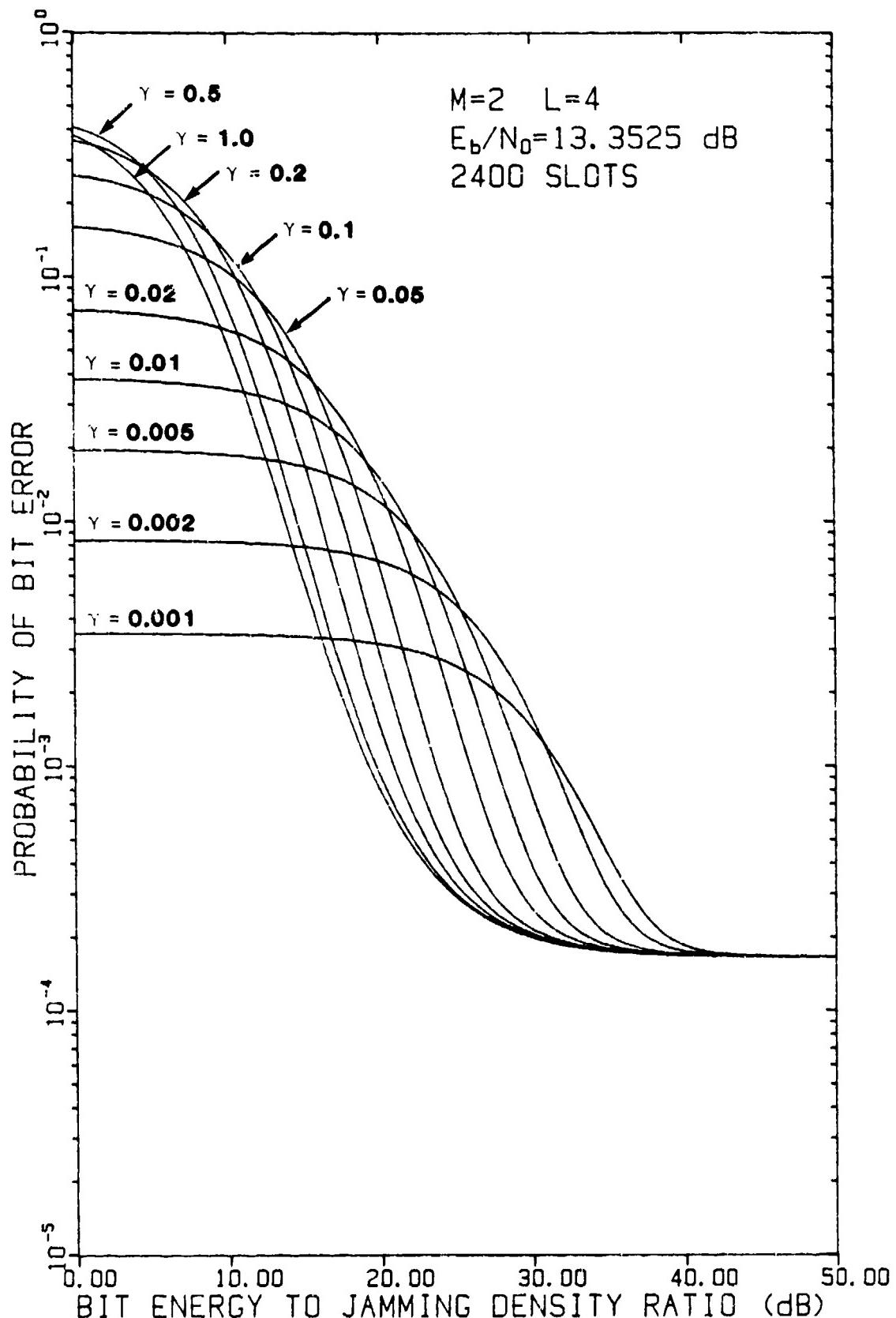


FIGURE 3.2-4 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH $M=2$ AND $L=4$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.3525 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L=1$) WITH JAMMING FRACTION γ AS A PARAMETER

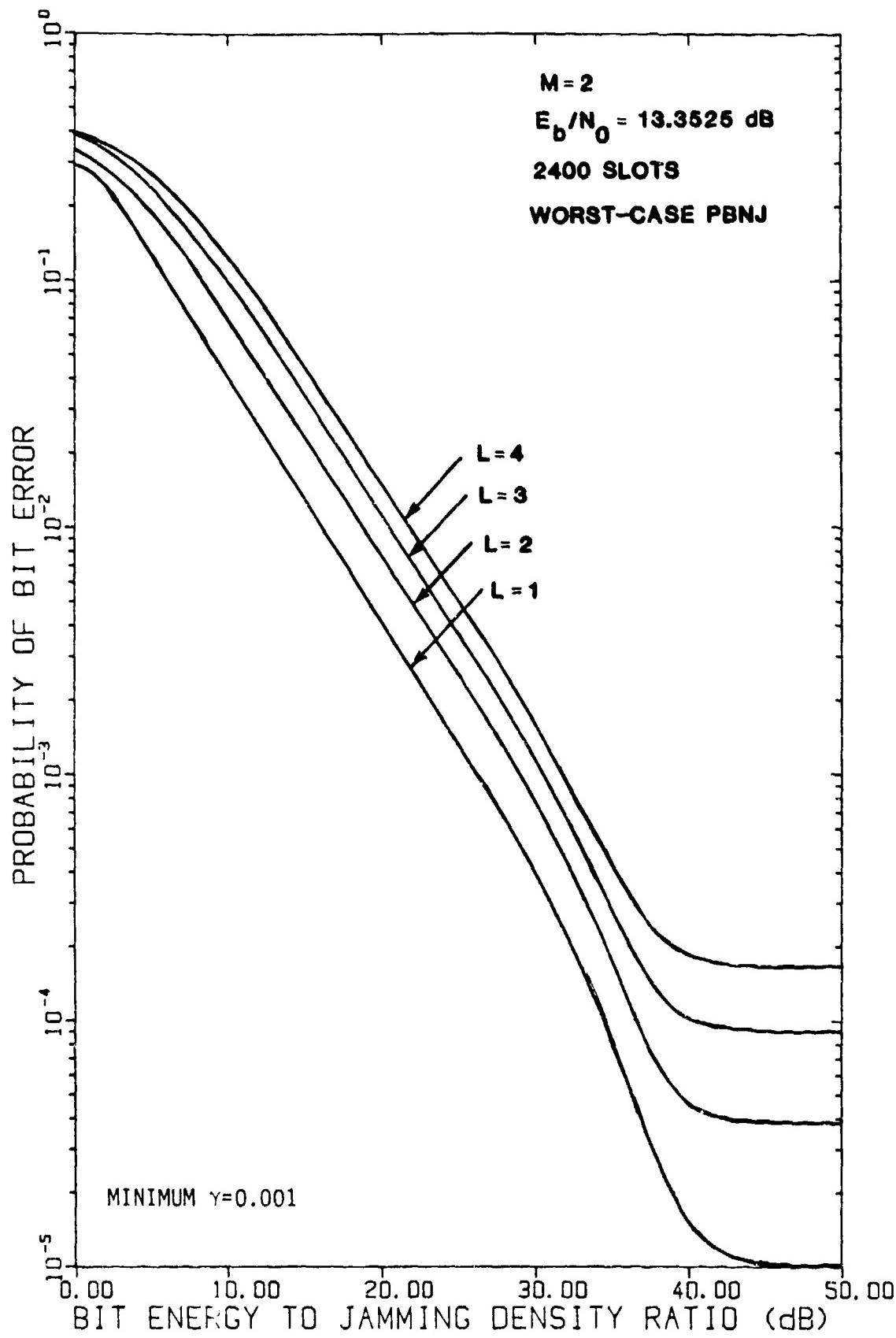


FIGURE 3.2-5 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER AND $M = 2$ WITH NUMBER OF HOPS/BIT AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING

Similar results for the case of $M=4$ are shown in Figures 3.2-6 through 3.2-9 for $L=1, 2, 3$, and 4 hops/symbol, respectively. Again we note that the jammer must carefully choose the proper partial-band fraction or risk reducing his effectiveness by more than an order of magnitude. We also observe that full-band jamming ($\gamma=1.0$) is not optimum until the jamming becomes very strong, i.e. $E_b/N_j < 0$ dB.

Figure 3.2-10 shows the envelope of the curves in Figures 3.2-6 through 3.2-9, which gives the performance in worst-case partial-band noise jamming. We note that increasing the number of hops per symbol consistently degrades the performance of the 4-ary system, just as it does for the binary system.

Finally, Figure 3.2-11 shows the worst-case partial-band noise jamming performance of the square-law receiver for $L=1$ hop/symbol and $M=2, 4$, and 8. We observe that for strong jamming increasing M from 2 to 4 provides a very small performance improvement; but further increase to $M=8$ degrades the performance. This behavior is similar to that of a block-hopping system in tone jamming [1, Section 8].

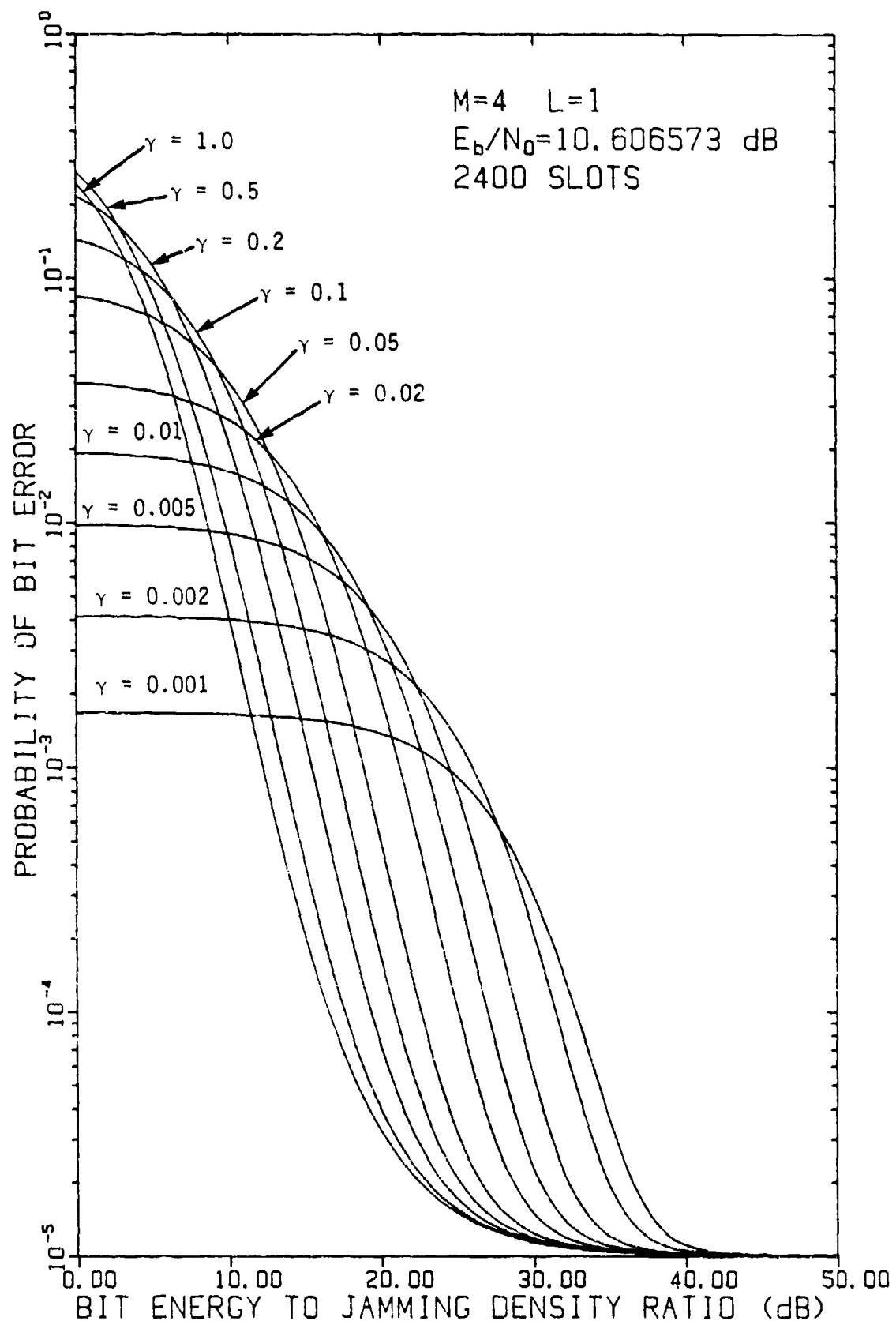


FIGURE 3.2-6 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH $M = 4$ AND $L = 1$ HOP/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L = 1$) WITH JAMMING FRACTION γ AS A PARAMETER

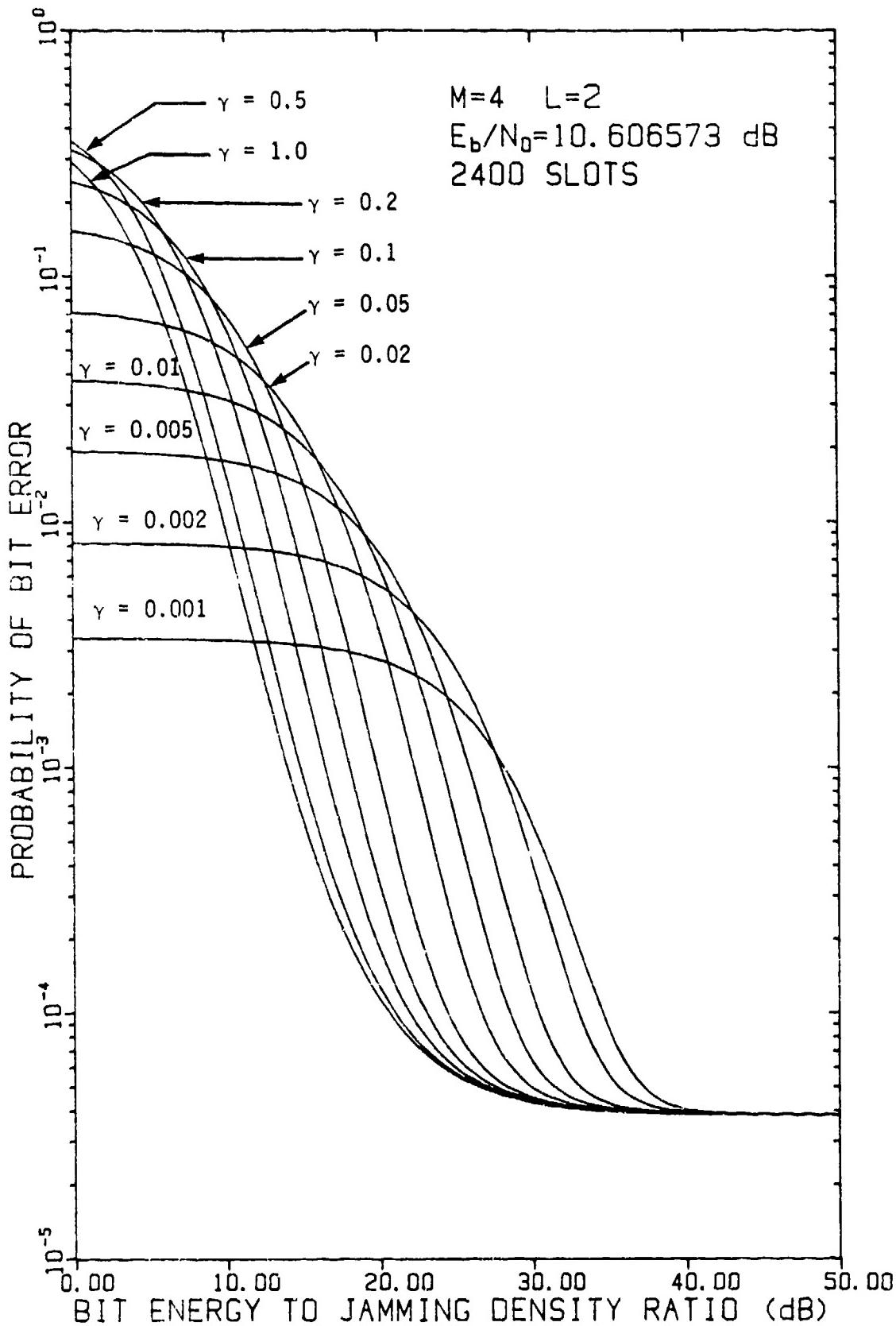


FIGURE 3.2-7 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH $M = 4$ AND $L = 2$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573 \text{ dB}$ (FOR 10^{-5} BER WITH-JAMMING WHEN $L = 1$) WITH JAMMING FRACTION γ AS A PARAMETER

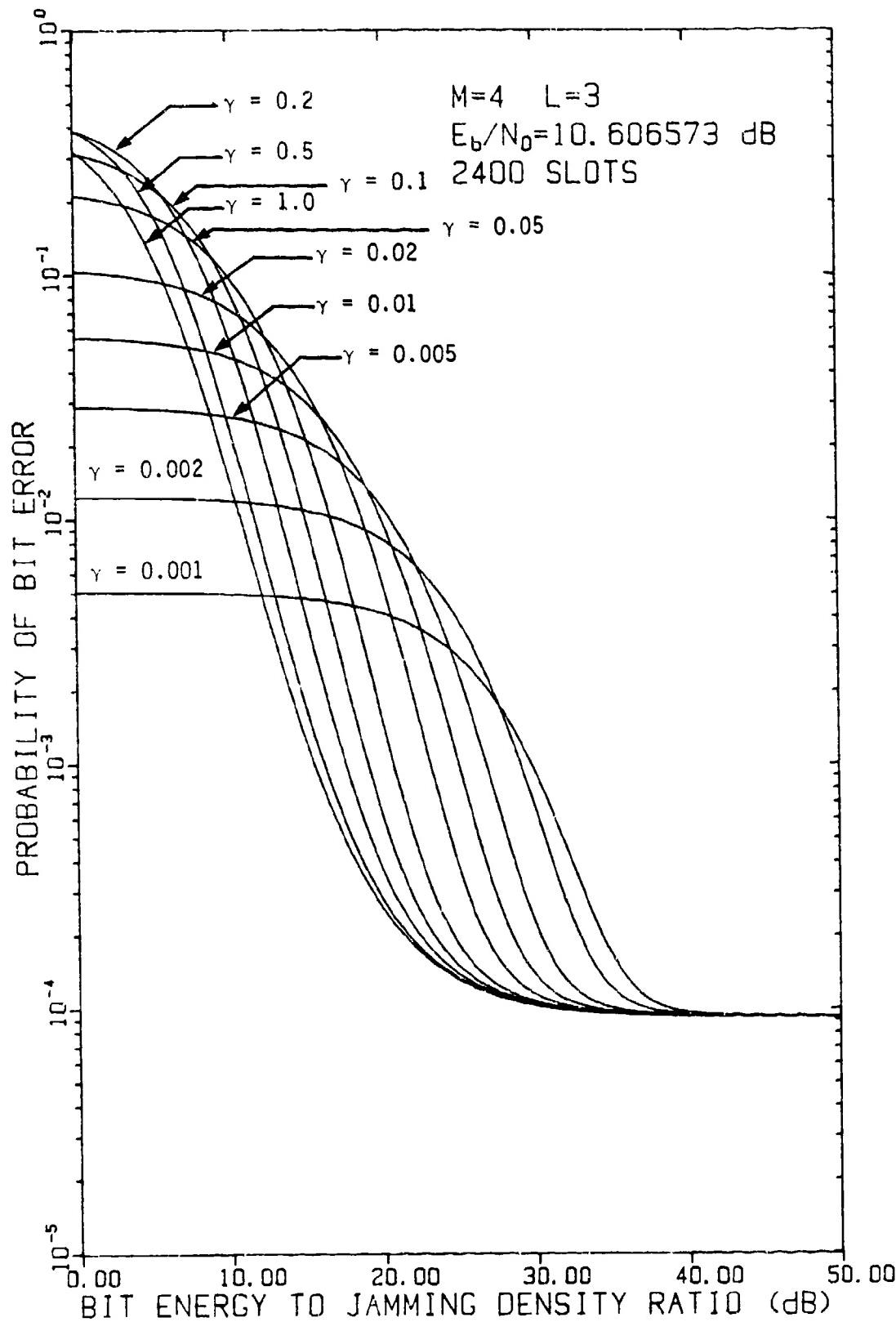


FIGURE 3.2-8 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH $M = 4$ AND $L = 3$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L = 1$) WITH JAMMING FRACTION γ AS A PARAMETER

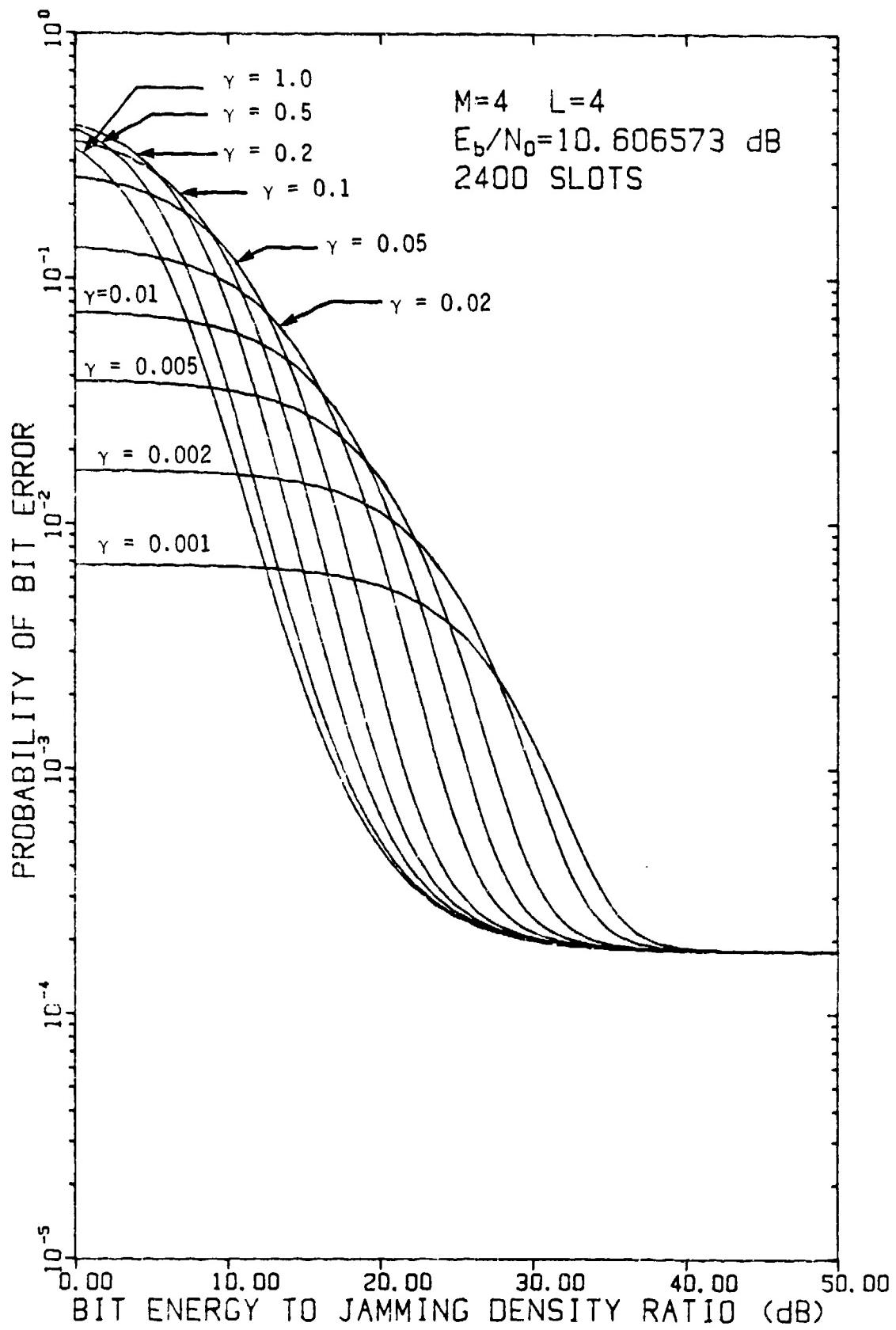


FIGURE 3.2-9 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH $M = 4$ AND $L = 4$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L = 1$) WITH JAMMING FRACTION γ AS A PARAMETER

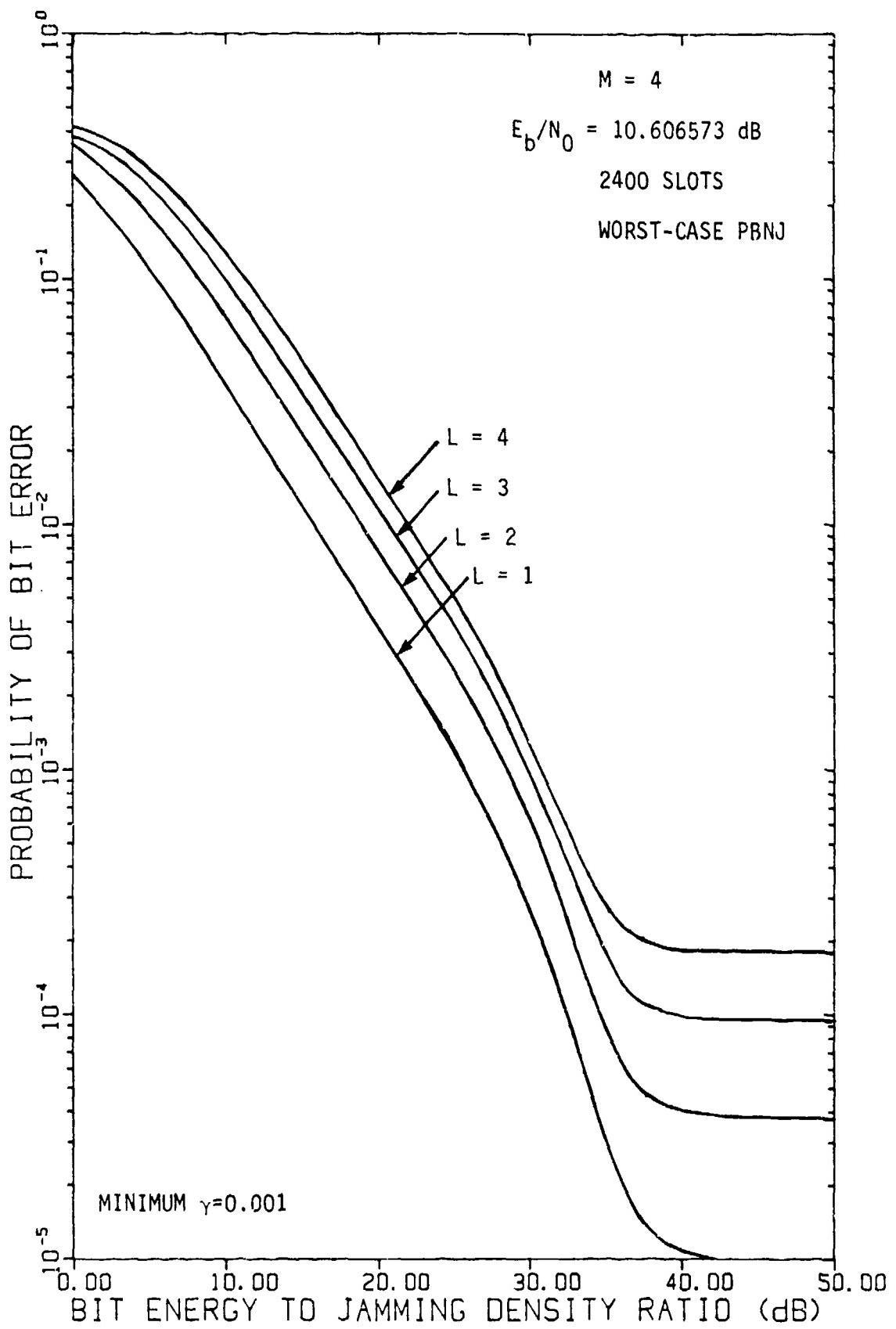


FIGURE 3.2-10 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER AND M = 4 WITH NUMBER OF HOPS/SYMBOL AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING

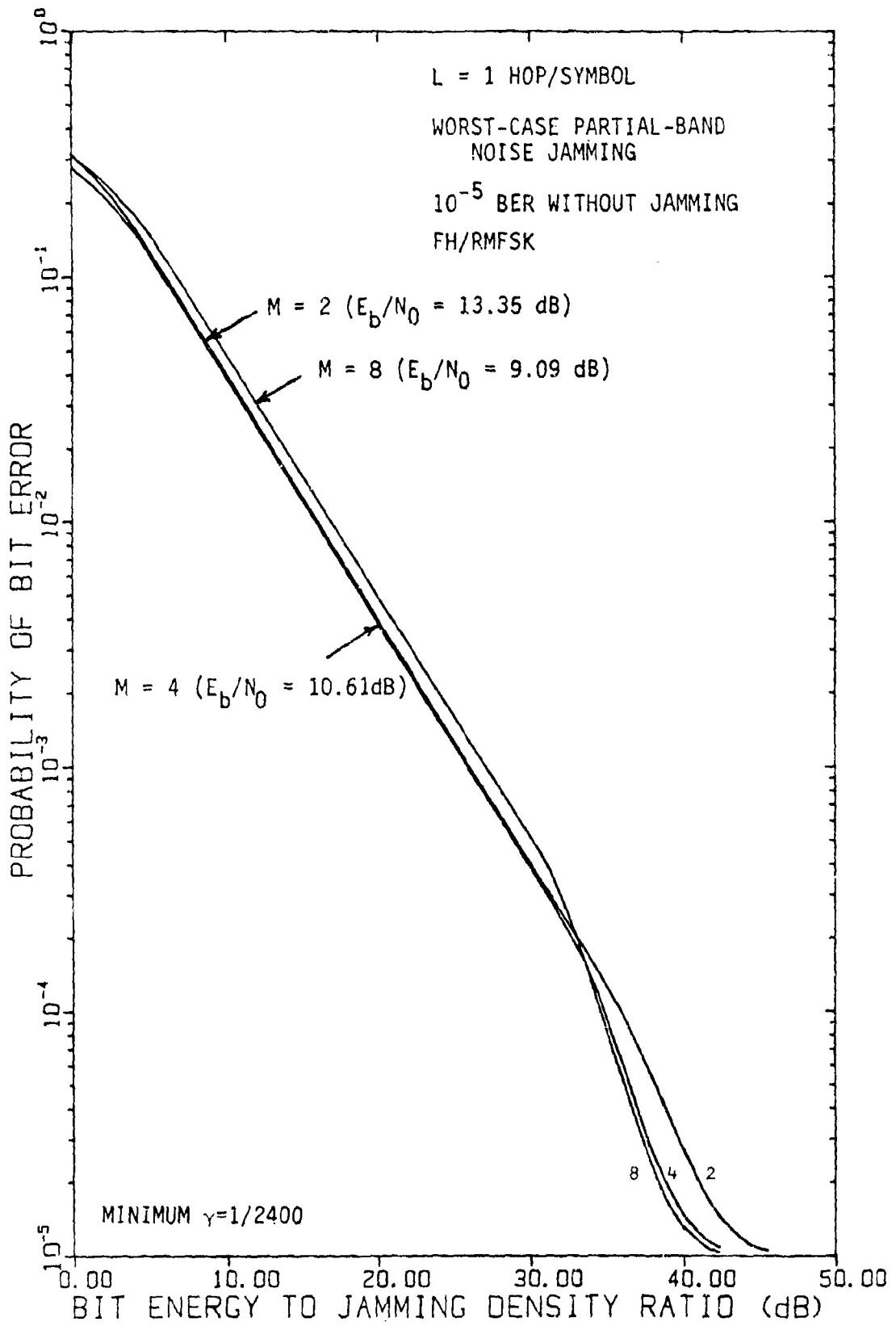


FIGURE 3.2-11 WORST-CASE PARTIAL-BAND NOISE JAMMING PERFORMANCE OF FH/RMFSK RECEIVERS FOR $L = 1$ HOP/SYMBOL AND $M = 2, 4, 8$ WHEN E_b/N_0 GIVES A 10^{-5} BER WITHOUT JAMMING

3.2.2 Hard-Decision Receiver

In this subsection we apply the explicit form of the error probability expression (2.4-14) to evaluate the symbol error probability, $P_s(e)$, for a square-law receiver with hard decisions. We consider M-ary cases of M=2, 4, and 8 with L values (hops per symbol) ranging from one through five. The worst-case or maximum probability of error is obtained by computing $P_s(e)$ upon varying the number of noise jammed hopping slots q, for an FH/RMFSK system comprised of 2400 hopping slots; i.e. a partial-band noise jamming (PBNJ) model.

We first present plots of numerical results for $P(e)$ versus the variable E_b/N_j with thermal noise (E_b/N_0) as a parameter. Practical values of E_b/N_0 were chosen for which the probability of error becomes 10^{-5} in the absence of jamming. These values are: 13.35247, 10.60657, and 9.09401 dB for M-ary signalling alphabets of M=2, 4, and 8 respectively. Corresponding performance plots are shown in Figures 3.2-12 to 3.2-14.

A comprehensive view of Figure 3.2-12 (M=2) reveals that all five of the L error curves could be grouped into three E_b/N_j regions of relative jamming strength: (1) below 11 dB (strong), (2) 11 dB to 28 dB (medium), and (3) beyond 28 dB (weak). Within the strong jamming region, we see a consistent $P(e)$ L-curve ordering of 4,5,2,3,1 and a 4,2,5,3,1 ranking in the weak region. The region defined as medium strength jamming exhibits "cross-overs" of the various L $P(e)$ curves. Similarly, for M=4 (Figure 3.2-13) and M=8 (Figure 3.2-14) this same three-region behavior exists as follows:

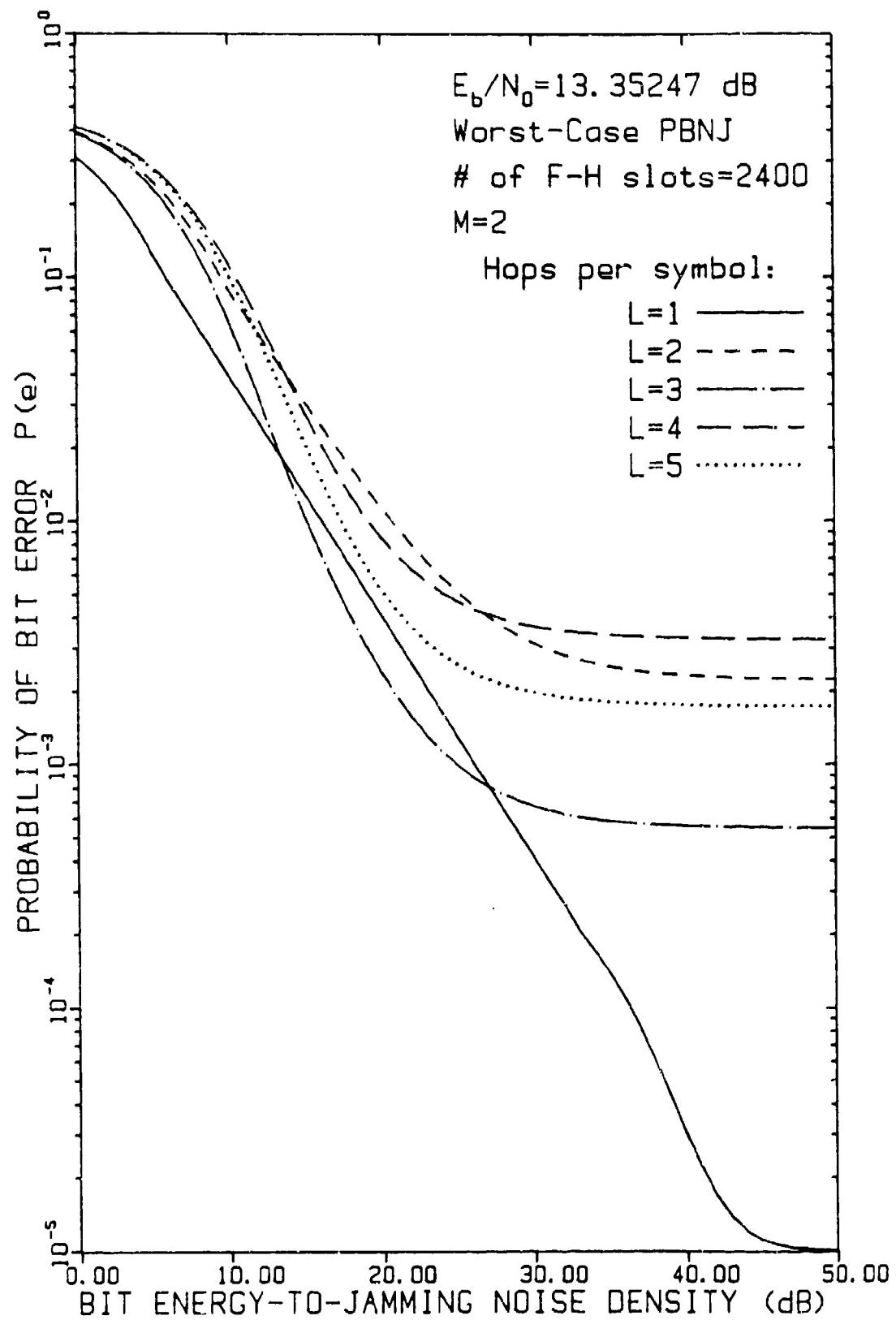


FIGURE 3.2-12 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR $M=2$ IN THE PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING AT $E_b/N_0 = 13.35247 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L=1$)

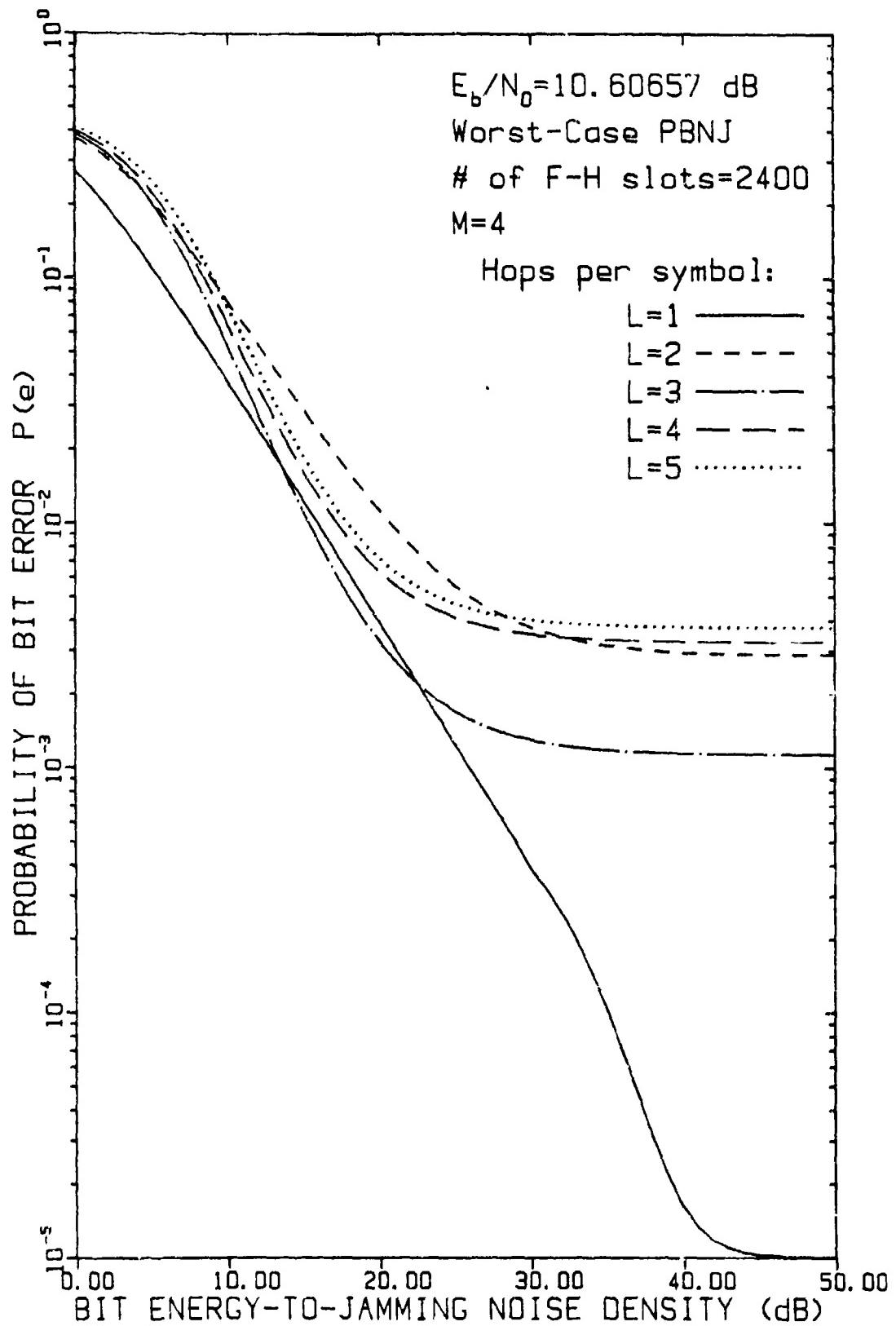


FIGURE 3.2-13 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO
 FOR SQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR $M=4$
 IN THE PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING AT
 $E_b/N_0 = 10.60657$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L=1$)

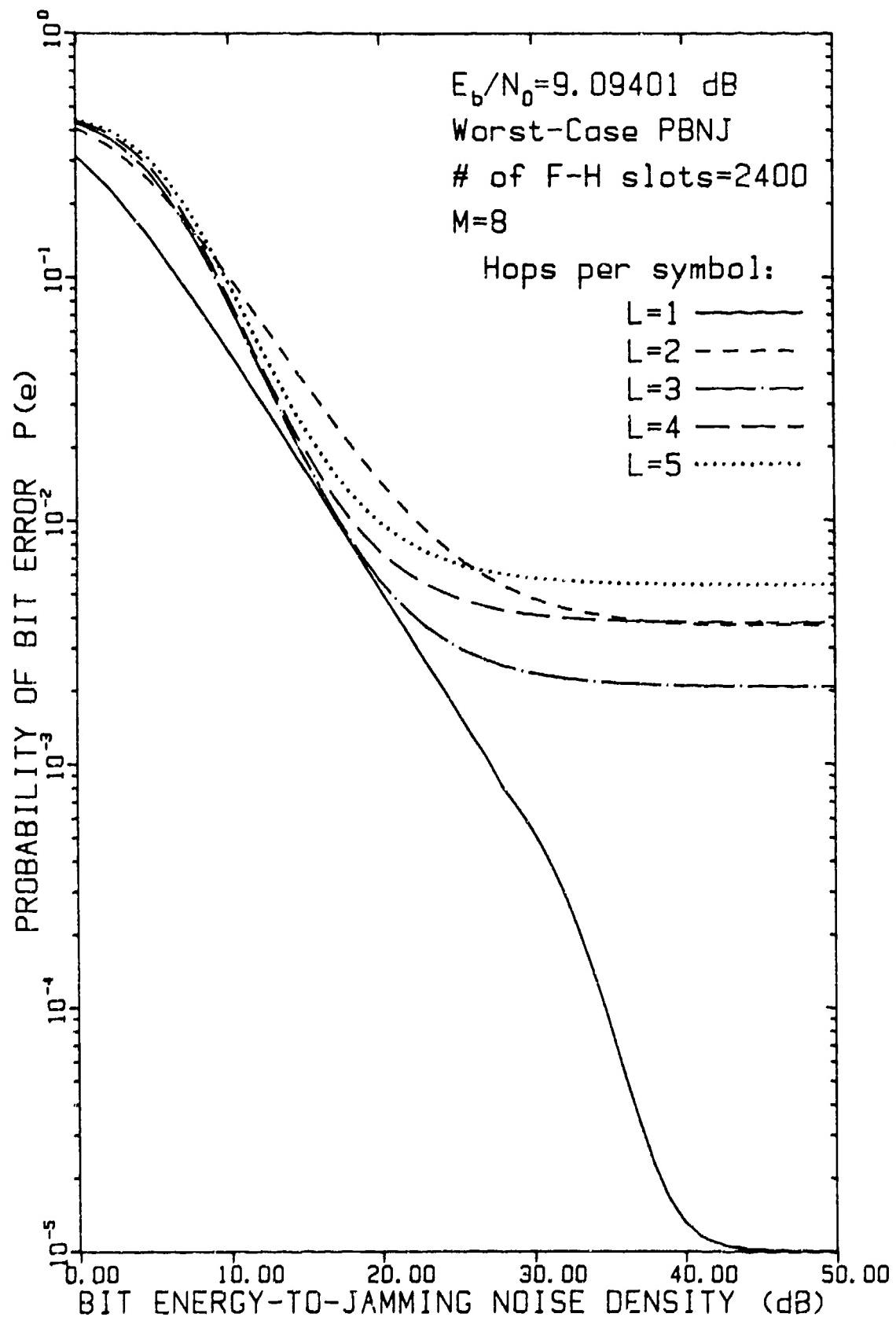


FIGURE 3.2-14 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR $M=8$ IN THE PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING AT $E_b/N_0 = 9.09401 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L=1$)

J. S. LEE ASSOCIATES, INC.

Relative Jamming Region of E_b/N_j (dB) Values

<u>M</u>	<u>Strong</u>	<u>Medium</u>	<u>Weak</u>
4	<4	4 to 32	>32
8	<6	6 to 38	>38 .

The ranking of the L P(e) curves for both M=4 and M=8 in the strong jamming region is 5,4,3,2,1 while for weak jamming a 5,4,2,3,1 ordering is observed.

It is plain to see that for the most part no diversity improvement is realized by the communicator with the exception of a portion of the L=3 P(e) curve in medium jamming for M=2 and M=4. This general behavior can be attributed to the dominance of the noncoherent combining loss existing for the stated thermal noise levels.

A somewhat different trend is noticed when the effect of thermal noise is minimized. Figures 3.2-15 through 3.2-17 show P(e) results for M=2, 4, and 8 respectively at E_b/N_0 levels of 20 dB. Clearly, the regions of strong and weak jamming are now quite discernable with a smaller crossover region (medium jamming) existing among the L P(e) curves. However, we do notice a ranking in the weak jamming areas that differs from those of the 10^{-5} parameter E_b/N_0 curves previously presented. Here it is easily seen that the hard-decision receiver is uniformly 3 dB better for L=1 than for L=2 as described by (2.4-15) for any value of M. Also, a form of diversity improvement is realized as L becomes greater than 2 for all M-ary cases at E_b/N_j values of more than about 12 dB.

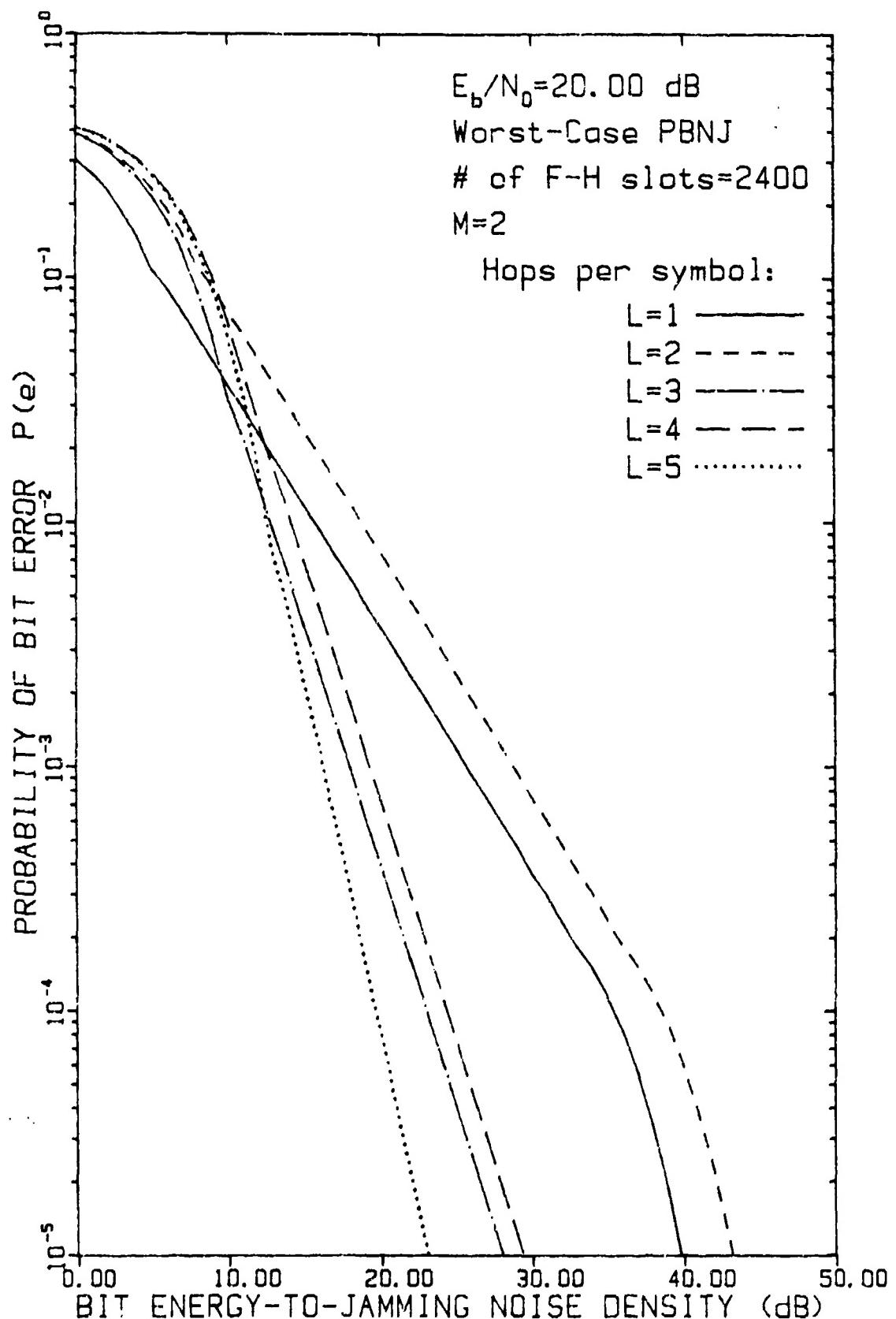


FIGURE 3.2-15 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR $M=2$ IN THE PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING AT $E_b/N_0 = 20 \text{ dB}$ (FOR MINIMIZATION OF THERMAL NOISE EFFECT)

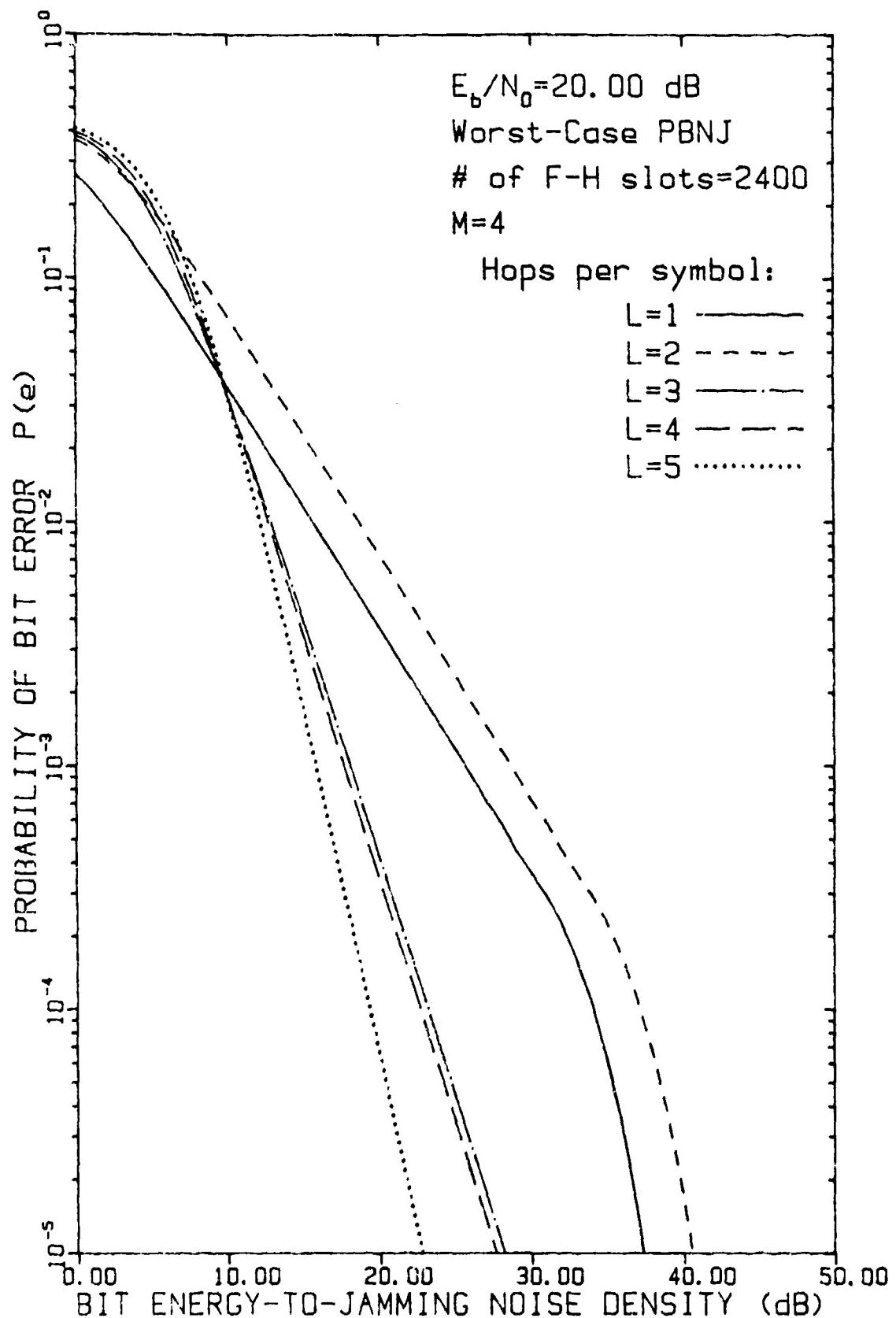


FIGURE 3.2-16 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR $M=4$ IN THE PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING AT $E_b/N_0 = 20 \text{ dB}$ (FOR MINIMIZATION OF THERMAL NOISE EFFECT)

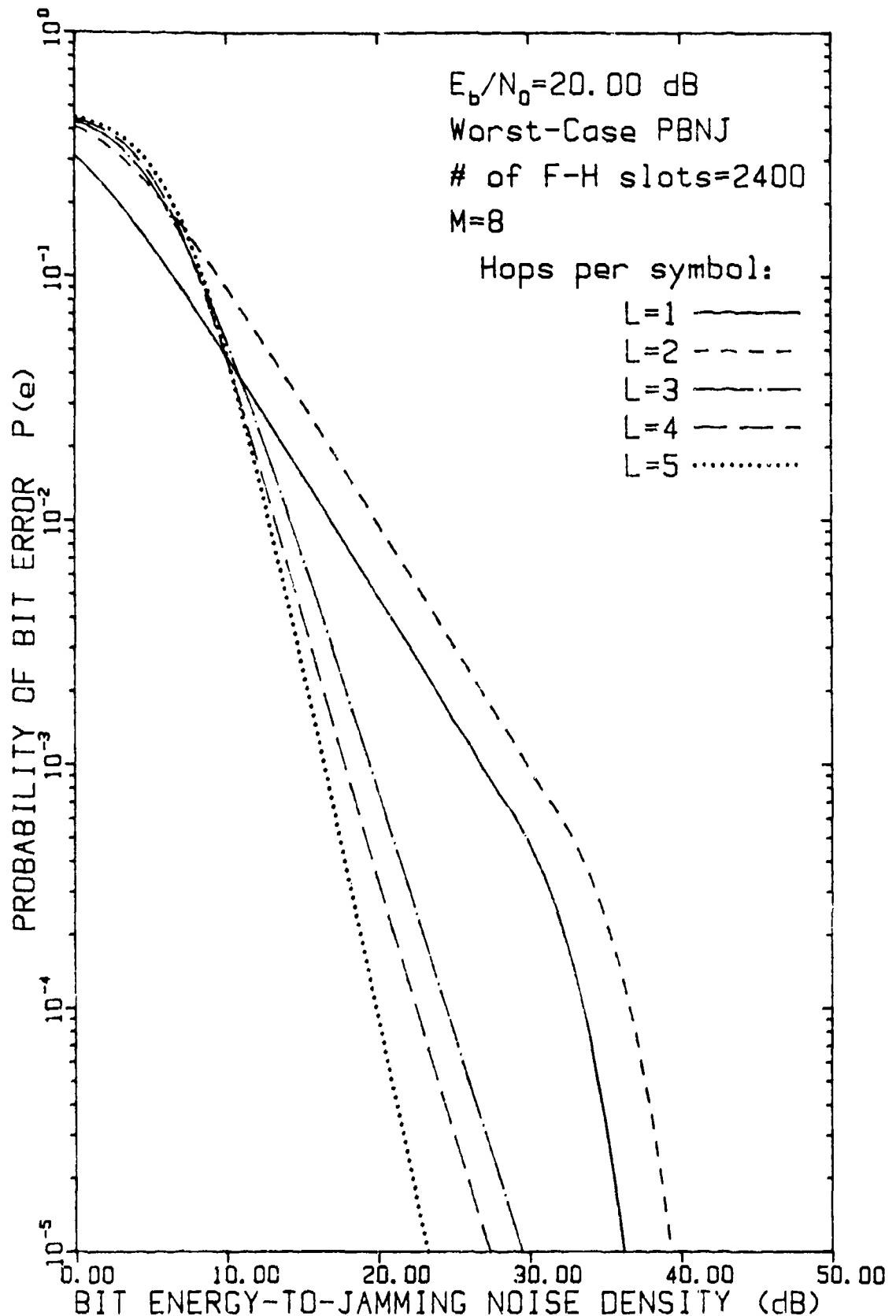


FIGURE 3.2-17 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR $M=8$ IN THE PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING AT $E_b/N_0 = 20 \text{ dB}$ (FOR MINIMIZATION OF THERMAL NOISE EFFECT)

J. S. LEE ASSOCIATES, INC.

The case of $M=2$ is an exception to this trend for $L=3$ and $L=4$ as explained in subsection 2.4.2.3 where $L=4$ is shown to be 1.25 dB worse than that for $L=3$.

Thus we conclude that when thermal noise is minimized, a form of diversity improvement does exist in all M cases for E_b/N_j values greater than around 12 dB for $L>2$ hops per symbol.

We now determine the optimum number of jammed slots (Q_{\max}) which yields the maximum probability of error for a given value of E_b/N_j . Figures 3.2-18 to 3.2-20 show such plots for case of $M=2, 4$, and 8 with L values ranging from one to five. It is seen that in all cases a definite ascending order of the $L Q_{\max}$ curves exists for increasing E_b/N_j values as is to be expected for worst-case jamming calculations. For example, in Figure 3.2-18 ($M=2$) we see that at a 30 dB E_b/N_j value, the Q_{\max} value is 2 for $L=1$ and over 200 for $L=5$. Also in Figure 3.2-18 we note the "plateau" effect for all the curves at Q_{\max} equal to 2400. Now the breakpoint at which each individual L -curve falls off from the "plateau" represents that E_b/N_j value for which full-band jamming ($\gamma=1.0$) will not cause maximum probability of error. We can also characterize each L -curve behavior of Figure 3.2-18 as per three definite regions with respect to the "slope" of each curve. These regions, in terms of Q_{\max} values, are: (1) 2400 to about 900, (2) 900 down to approximately 20, and (3) below 20. Note that distinguishable breaks in the curves below 20 are due to the smaller quantized values of Q becoming more discernable for the lower values of the logarithmic Q_{\max} scale.

With regard to Figures 3.2-19 ($M=4$) and 3.3-20 ($M=8$), we see an asymptotic merging of the L -curves within the region of approximately $Q_{\max} = 800$ to 2400. Below a Q_{\max} of about 800 we have two more noticeable regions

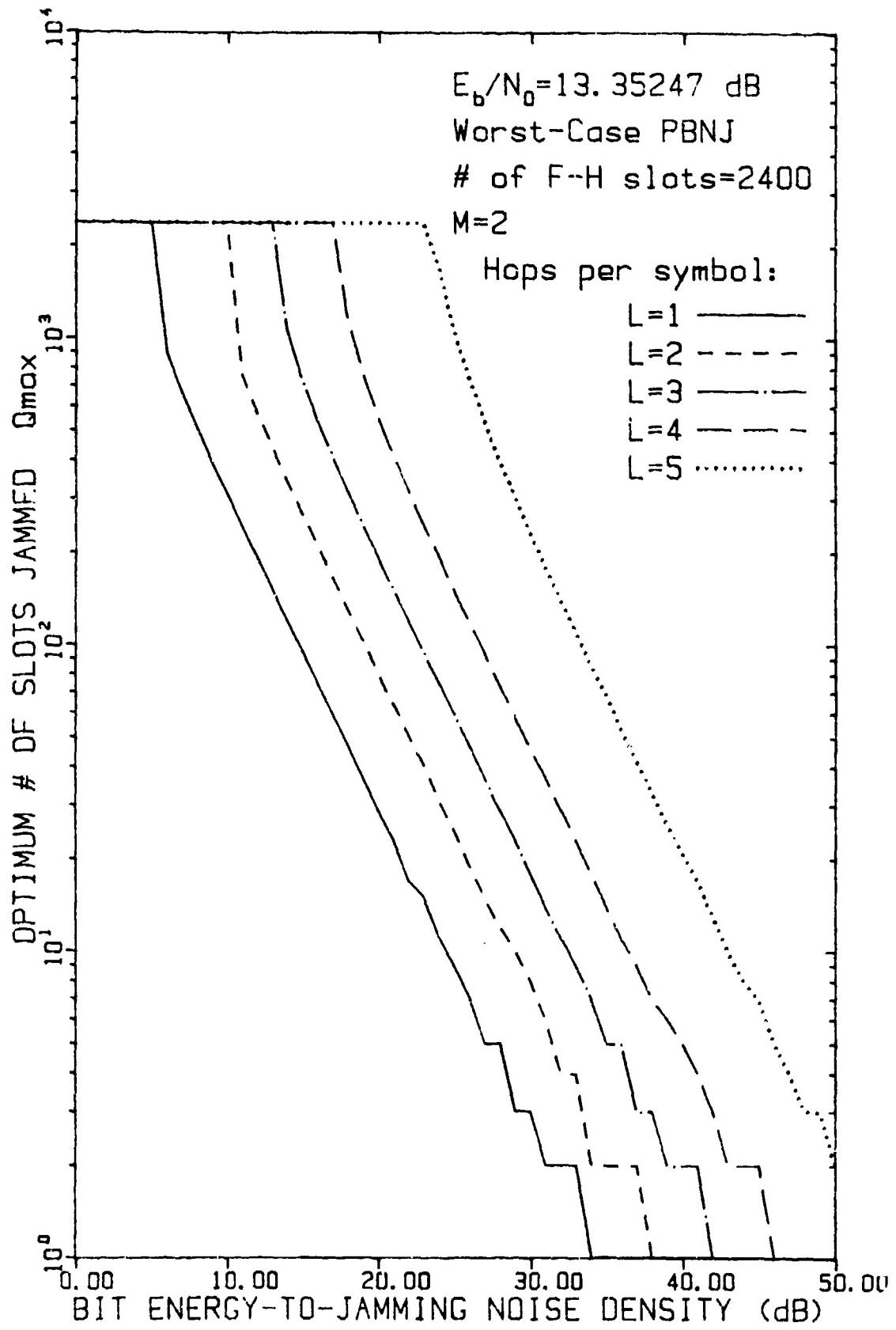


FIGURE 3.2-18 OPTIMUM NUMBER OF HOPPING SLOTS JAMMED (Q_{\max}) THAT PRODUCES $P(e)_{\max}$ VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR THE SQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR $M=2$ AT $E_b/N_0 = 13.35246 \text{ dB}$ (FOR 10^{-5} BER WHEN $L=1$)

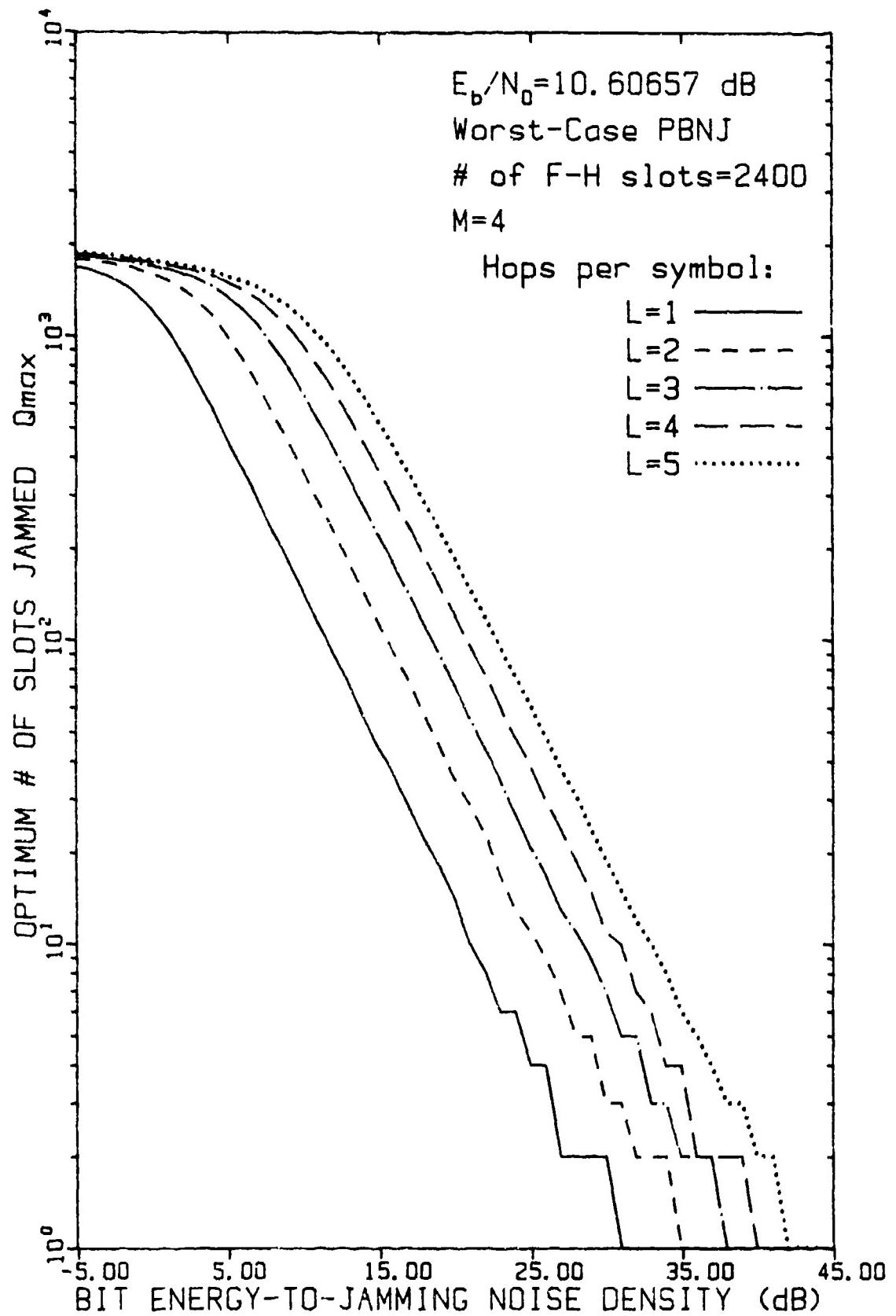


FIGURE 3.2-19 OPTIMUM NUMBER OF HOPPING SLOTS JAMMED (Q_{\max}) VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR THE SQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR $M=4$ AT $E_b/N_0 = 10.60657 \text{ dB}$ (FOR 10^{-5} BER WHEN $L=1$)

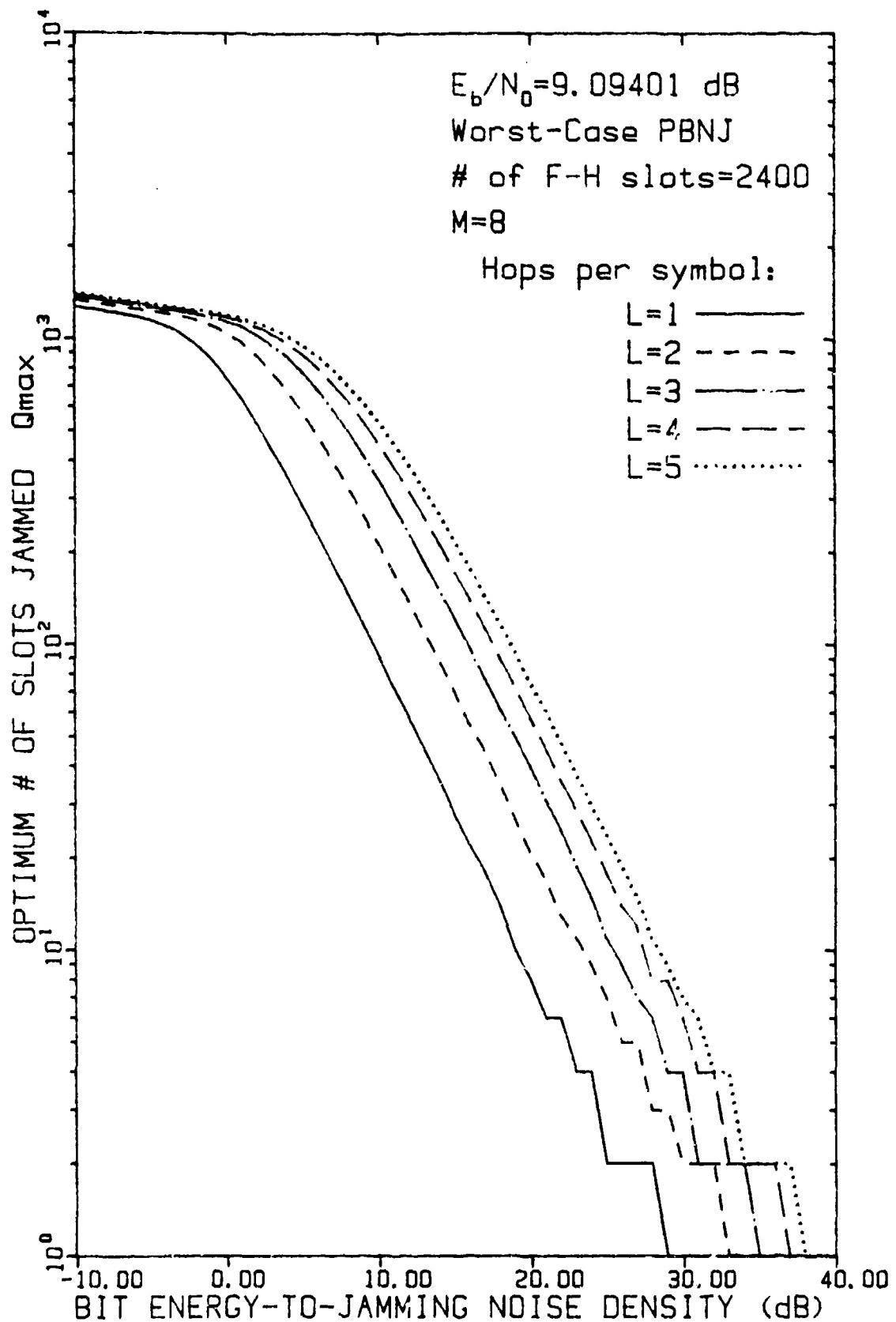


FIGURE 3.2-20 OPTIMUM NUMBER OF HOPPING SLOTS JAMMED (Q_{\max}) THAT PRODUCES $P(e)_{\max}$ VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR THE SQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR $M=8$ AT $E_b/N_0 = 9.09401 \text{ dB}$ (FOR 10^{-5} BER WHEN $L=1$)

J. S. LEE ASSOCIATES, INC.

similar to those in Figure 3.2-18. Additionally, we see that this asymptotic merging has yet to reach a full-band value of $Q_{\max} = 2400$ for the minimum E_b/N_J value utilized in the calculations. Further computations (not shown) for lower values of E_b/N_J reveal the following points at which the $L=1$ curve breaks from the $Q_{\max} = 2400$ ($\gamma = 1.0$) value: $M=4$ at -149.0 dB, $M=8$ at -150.0 dB. Hence, in these cases, full-band jamming would only be optimum for a very large amount of available jamming power.

A final point of interest is shown in Figure 3.2-21 with respect to minimization of the thermal noise component. Here we see that for the binary ($M=2$) case, increasing the E_b/N_0 value over that used in Figure 3.2-18 causes the $L=5$ curve to move around 10 dB (E_b/N_J) lower while the $L=1$ curve decreases only about 1 dB.

The results indicate that the hard symbol decision receiver can be considered an ECCM receiver (for sufficiently high E_b/N_0), while the linear combining receiver cannot. Therefore, it is not diversity as such that yields an ECCM effect, but the combining technique. Hard decisions (~ form of repetition coding) in effect limit the jamming effects on a given hop to that hop, whereas with linear combining a single, strongly jammed hop can dominate the soft decision.

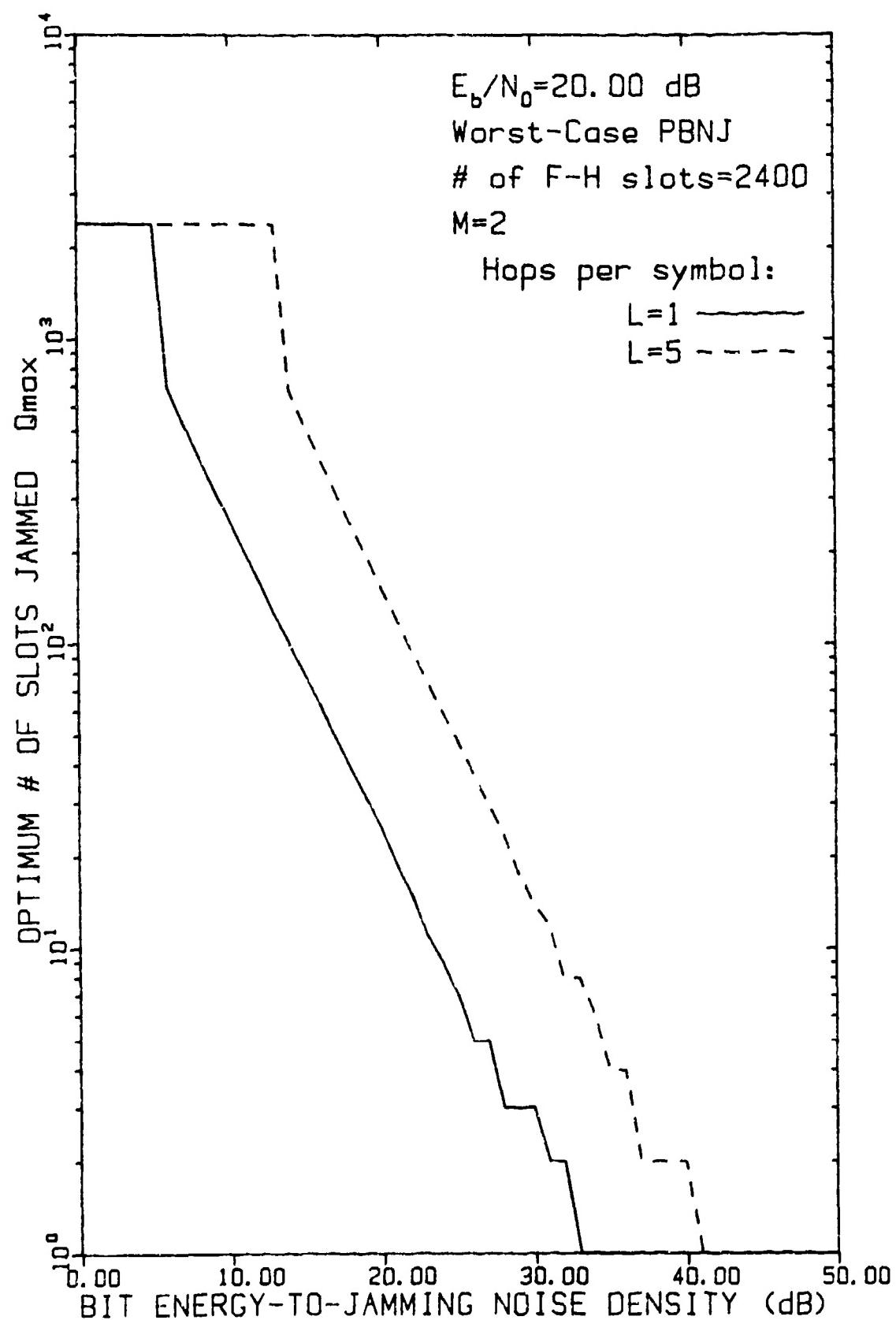


FIGURE 3.2-21 OPTIMUM NUMBER OF HOPPING SLOTS JAMMED (Q_{\max}) THAT PRODUCES $P(e)_{\max}$ VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR THE SQUARE-LAW COMBINING RECEIVER WITH HARD DECISIONS FOR $M=2$ AND $L=1,5$ AT $E_b/N_0=20$ dB (FOR MINIMIZATION OF THERMAL NOISE)

J. S. LEE ASSOCIATES, INC.

4.0 FH/RMFSK PERFORMANCE USING SQUARE-LAW AGC RECEIVER

In this section we consider the case of the generic receiver shown in Figure 2.2-1 when the envelope samples are processed using the function

$$f(x_{mk}) = x_{mk}^2 \cdot w_{mk} = z_{mk} . \quad (4.0-1)$$

That is, the decision statistics $\{z_m\}$ are weighted sums of samples of the squared envelope in each channel over multiple (L) hops.

For conventional FH/MFSK, where the symbol frequency slots are hopped together, it was assumed in [1] that all the slots are jammed or all the slots are not jammed on a given hop, and the weights were taken to be

$$w_{mk} = 1/\sigma_k^2 = \begin{cases} 1/\sigma_N^2, & \text{hop not jammed} \\ 1/\sigma_J^2, & \text{hop jammed.} \end{cases} \quad (4.0-2)$$

This weighting or normalization scheme was predicated on use of a separate channel, or perhaps a look-ahead scheme, to measure the noise power (perfectly) on each hop. The effect of the weighting is to de-emphasize the jammed hops in the summations

$$z_m = \sum_{k=1}^L z_{mk} , \quad (4.0-3)$$

and therefore to mitigate the effect of the jamming on the symbol decision.

For FH/RMFSK, in general the different symbol frequency slots are independently jammed or not jammed when the system bandwidth contains power from a partial-band noise jammer. In Section 4.1, we discuss several normalization schemes of the AGC (adaptive gain control) type. The performances of two of these schemes are analyzed in Sections 4.2 and 4.3. We also consider the effect of hard-limiting the hop statistics $\{z_{mk}\}$ prior to combining.

J. S. LEE ASSOCIATES, INC.

4.1 POSSIBLE AGC WEIGHTING SCHEMES FOR FH/RMFSK

The squared envelope samples in the M receiver channels on a given hop are weighted chi-squared random variables:

$$\begin{aligned} x_{1k}^2 &\sim \sigma_{1k}^2 \chi^2(2; \lambda_{1k} = 2S/\sigma_{1k}^2) \\ &= (\sigma_N^2 + v_{1k} \sigma_J^2) \chi^2[2; 2S/(\sigma_N^2 + v_{1k} \sigma_J^2)] \end{aligned} \quad (4.1-1a)$$

in the signal channel, and

$$x_{mk}^2 \sim \sigma_{mk}^2 \chi^2(2) = (\sigma_N^2 + v_{mk} \sigma_J^2) \chi^2(2), m > 1, \quad (4.1-1b)$$

in the non-signal channels, where

$$v_{mk} = \begin{cases} 1, & \text{channel } m \text{ jammed on hop } k \\ 0, & \text{channel } m \text{ not jammed on hop } k. \end{cases} \quad (4.1-2)$$

We shall consider three approaches to AGC normalization, as illustrated in Figure 4.1-1:

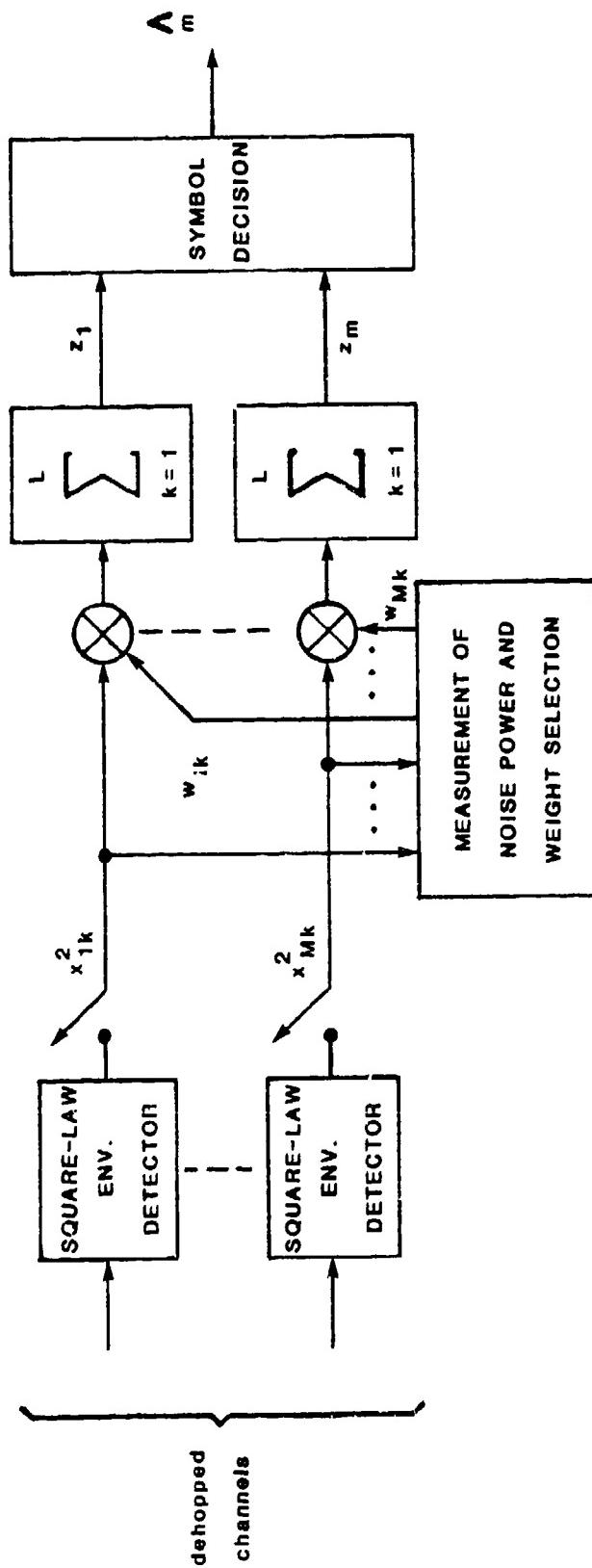
(a) Measurement of noise power in the M de-hopped channels on each hop and normalization (division) by the average received noise power (variance) in these channels.

(b) Individual measurement and normalization of each of the M channels on each hop.

(c) Normalization of the M channels by the same amount, depending on whether one or more of the channels are jammed, or none are jammed.

4.1.1 Average AGC Scheme.

An ideal measurement of average noise power in the M channels would yield the weights



Weighting schemes:

(a) Average AGC approach:

$$w_{mk} = \left[\frac{1}{2} E \left\{ \sum_{m=1}^M x_{mk}^2 - S \right\} \right]^{-1} = \left[\frac{1}{M} \sum_{m=1}^M \sigma_{mk}^2 \right]^{-1} = w_k$$

(b) Individual-channel AGC approach: $w_{mk} = (\sigma_{mk}^2)^{-1}$

$$(c) \text{ Any-channel-jammed AGC approach: } w_{mk} = w_k = \begin{cases} \left(\sigma_N^2 \right)^{-1} & \text{no channels jammed} \\ \left(\sigma_T^2 \right)^{-1} & \text{one or more channels jammed} \end{cases}$$

FIGURE 4.1-1 POSSIBLE AGC NORMALIZATION SCHEMES

$$w_{mk} \equiv w_k = \left(\frac{1}{M} \sum_{m=1}^M q_{mk}^2 \right)^{-1}$$

$$= \left(\sigma_N^2 + \sigma_J^2 \sum_{m=1}^M v_{mk}/M \right)^{-1}. \quad (4.1-3)$$

There are M possible values to these weights. After normalization, the $\{z_{mk}\}$ become

$$z_{1k} \sim w_k \sigma_{1k}^2 \chi^2(2; 2S/\sigma_{1k}^2) \quad (4.1-4a)$$

$$z_{mk} \sim w_k \sigma_{mk}^2 \chi^2(2), \quad m > 1. \quad (4.1-4b)$$

The effective weights $w_k \equiv w_k \sigma_{mk}^2$ on the chi-squared variables in a given channel can take $2(M-1)$ values. Therefore the decision statistics for this normalization scheme have the form

$$z_m \sim \sum_{k=1}^L w_k \chi^2(2; \lambda_{mk}). \quad (4.1-5)$$

The distribution of sums of non-equally weighted chi-squared random variables is extremely difficult to compute. For this reason, it is not feasible to consider calculation of the error performance using such a weighting scheme.

4.1.2 Individual Channel AGC Scheme.

Ideal measurements of noise power in each of the M channels would yield the weights

$$w_{mk} = (\sigma_{mk}^2)^{-1} \quad (4.1-6)$$

and the decision statistics

$$z_1 = \sum_{k=1}^L \chi^2(2; \lambda_{1k}) = \chi^2(2L; \lambda_1) \quad (4.1-7a)$$

J. S. LEE ASSOCIATES, INC.

in the signal channel, where

$$\lambda_1 = \sum_{k=1}^L \lambda_{1k} = (L - \ell_1) \cdot 2S/\sigma_N^2 + \ell_1 \cdot 2S/\sigma_T^2 , \quad (4.1-7b)$$

and in the non-signal channels,

$$z_m = \sum_{k=1}^L x^2(2) = x^2(2L) . \quad (4.1-7c)$$

Thus whatever else the merits of this normalization scheme may be, it yields decision statistics which are purely chi-squared random variables with $2L$ degrees of freedom. In fact, from (4.1-7) we observe that the discernable jamming events are characterized solely by the number of hops jammed in the signal channel, ℓ_1 .

4.1.3 Any-Channel-Jammed AGC Scheme.

This scheme takes the approach that if any of the channels on a given hop is jammed, then all the channels are normalized by $\sigma_T^2 = \sigma_N^2 + \sigma_J^2$; otherwise they are all normalized by σ_N^2 . Expressed mathematically, the weights are

$$w_{mk} \equiv w_k = \begin{cases} (\sigma_N^2)^{-1}, & \sum_{m=1}^M v_{mk} = 0 \\ (\sigma_T^2)^{-1}, & \text{otherwise.} \end{cases} \quad (4.1-8)$$

The result of this approach is that the hop statistics fall into three categories:

- (a) channel not jammed, normalized by σ_N^2
- (b) channel not jammed, normalized by σ_T^2
- (c) channel jammed, normalized by σ_T^2 .

J. S. LEE ASSOCIATES, INC.

If we define ℓ_0 as the number of hops on which at least one of the channels is jammed, we find that in channel m ($m=1, 2, \dots, M$) there would be

- * $(L-\ell_0)$ hops with noise power σ_N^2 normalized by σ_T^2
- * $(\ell_0-\ell_m)$ hops with noise power σ_N^2 normalized by σ_T^2
- * ℓ_m hops with noise power σ_T^2 normalized by σ_T^2 .

Therefore, for a given ℓ_0 and jamming event vector $\underline{\ell} = (\ell_1, \ell_2, \dots, \ell_M)$, the decision statistics would have the following distributions:

$$\begin{aligned} z_1 &\sim \chi^2[2(L-\ell_0); 2(L-\ell_0)S/\sigma_N^2] + (\sigma_N^2/\sigma_T^2)\chi^2[2(\ell_0-\ell_1); 2(\ell_0-\ell_1)S/\sigma_N^2] \\ &\quad + \chi^2[2\ell_1; 2\ell_1S/\sigma_T^2] \\ &= \chi^2[2(L+\ell_1-\ell_0); 2(L-\ell_0)S/\sigma_N^2 + 2\ell_1S/\sigma_T^2] \\ &\quad + K^{-1}\chi^2[2(\ell_0-\ell_1); 2(\ell_0-\ell_1)S/\sigma_N^2], \quad \ell_1 \neq \ell_0; \end{aligned} \quad (4.1-9a)$$

and

$$\begin{aligned} z_m &\sim \chi^2[2(L-\ell_0)] + (\sigma_N^2/\sigma_T^2)\chi^2[2(\ell_0-\ell_m)] + \chi^2(2\ell_m) \\ &= \chi^2[2(L+\ell_m-\ell_0)] + K^{-1}\chi^2[2(\ell_0-\ell_m)], \quad m > 1; \quad \ell_m \neq \ell_0. \end{aligned} \quad (4.1-9b)$$

When $\ell_m = \ell_0$ the case of noise power σ_N^2 normalized by σ_T^2 does not occur and the distributions are:

$$z_1 \sim \chi^2[2L; 2(L-\ell_0)S/\sigma_N^2 + 2\ell_0S/\sigma_T^2], \quad \ell_1 = \ell_0; \quad (4.1-9c)$$

and

$$z_m \sim \chi^2(2L), \quad \ell_m = \ell_0. \quad (4.1-9d)$$

As in Section 3, we use $K \triangleq \sigma_T^2/\sigma_N^2 > 1$. We see from (4.1-9) that the decision statistics are in general sums of two unequally weighted chi-squared random variables. Analysis of this distribution is difficult but has been accomplished previously, in Section 3. There is the additional complexity, however, that the jamming events now must be described by an additional parameter: ℓ_0 , the number of hops with at least one channel jammed. This task can be achieved as shown below.

J. S. LEE ASSOCIATES, INC.

4.2 ANALYSIS OF FH/RMFSK PERFORMANCE USING INDIVIDUAL CHANNEL AGC SCHEME

Now we obtain the probability of bit error for the FH/RMFSK receiver using the individual channel AGC normalization scheme.

4.2.1 Conditional Probability Of Error.

The probability of a symbol error, given a jamming event described by $\underline{\ell} = (\ell_1, \ell_2, \dots, \ell_M)$, is

$$\begin{aligned} P_s(e|\underline{\ell}) &= 1 - P_s(C|\underline{\ell}) \\ &= 1 - \int_0^\infty d\alpha p_{z_1}(\alpha) \prod_{m=2}^M \int_0^\alpha d\beta_m p_{z_m}(\beta_m). \end{aligned} \quad (4.2-1)$$

From (4.1-7) we observe that the non-signal channel decision statistics $\{z_m, m>1\}$ are identically distributed as chi-squared random variables with $2L$ degrees of freedom. Thus

$$\begin{aligned} \int_0^\alpha d\beta_m p_{z_m}(\beta_m) &= \int_0^\alpha d\beta p_{z_2}(\beta), \quad m = 2, 3, \dots, M \\ &= 1 - \frac{I(L; \alpha/2)}{\Gamma(L)} \end{aligned} \quad (4.2-2a)$$

$$= 1 - e^{-\alpha/2} \sum_{r=0}^{L-1} \frac{(\alpha/2)^r}{r!}, \quad (4.2-2b)$$

and

$$\begin{aligned} P_s(C|\underline{\ell}, \alpha) &\triangleq \prod_{m=2}^M \int_0^\alpha d\beta_m p_{z_m}(\beta_m) = \left[1 - e^{-\alpha/2} \sum_{r=0}^{L-1} \frac{(\alpha/2)^r}{r!} \right]^{M-1} \\ &= \sum_{k=0}^{M-1} \binom{M-1}{k} (-1)^k e^{-k\alpha/2} \left[\sum_{r=0}^{L-1} \frac{(\alpha/2)^r}{r!} \right]^k. \end{aligned} \quad (4.2-3)$$

From Section 3, equation (3.1-14), we find that

$$P_s(C|L, \alpha) = \sum_{k=0}^{M-1} \binom{M-1}{k} (-1)^k e^{-k\alpha/2} \sum_{r=0}^{k(L-1)} \frac{C(r,k)}{r!} \left(\frac{\alpha}{2}\right)^r, \quad (4.2-4a)$$

where

$$\begin{aligned} C(0,k) &= 1 \\ C(r,k) &= \frac{1}{r} \sum_{n=1}^{\min(r,L-1)} \binom{r}{n} [(k+1)n - r] C(r-n,k), \\ &\quad r>0, L>2. \end{aligned} \quad (4.2-4b)$$

Substitution in (4.2-1) yields

$$P_s(e|L) = 1 - \sum_{k=0}^{M-1} \binom{M-1}{k} (-1)^k \sum_{r=0}^{k(L-1)} \frac{C(r,k)}{r!} \int_0^\infty d\alpha P_{z_1}(\alpha) e^{-k\alpha/2} (\alpha/2)^r. \quad (4.2-5)$$

From (4.1-7) and (3.1-9) the pdf for z_1 is

$$P_{z_1}(\alpha) = \frac{1}{2} e^{-(\alpha+\lambda_1)/2} \left(\frac{\alpha}{\lambda_1}\right)^{(L-1)/2} I_{L-1}\left(\sqrt{\alpha\lambda_1}\right), \quad (4.2-6)$$

with $\lambda_1 = 2(L-\ell_1)\rho_N + 2\ell_1\rho_T$. Thus the required integral in (4.2-5) is

$$\begin{aligned} &\left(\frac{1}{1+k}\right)^{r+L} \int_0^\infty dx \left(\frac{x}{2}\right)^r \exp\left(-\frac{k\lambda_1/2}{k+1}\right) P_{x^2}(x; 2L, \lambda = \frac{\lambda_1}{1+k}) \\ &= \left(\frac{1}{1+k}\right)^{r+L} \exp\left(-\frac{k\lambda_1/2}{k+1}\right) r! \quad \varrho_r^{L-1}\left(\frac{\lambda_1/2}{k+1}\right), \end{aligned} \quad (4.2-7)$$

giving the error probability

$$P_s(e|\underline{\lambda}) = \sum_{k=1}^{M-1} \binom{M-1}{k} \frac{(-1)^{k+1}}{(1+k)^L} \sum_{r=0}^{k(L-1)} \frac{C(r,k)}{(1+k)^r} \exp\left(-\frac{k\lambda_1/2}{k+1}\right) \rho_r^{L-1}\left(\frac{-\lambda_1/2}{1+k}\right) \quad (4.2-8)$$

$$\equiv P_s(e|\lambda_1).$$

4.2.2 Total Error Probability.

Since the symbol error probability depends only on whether the signal channel is jammed, the total bit error probability is

$$P_b(e) = \sum_{\ell_1=0}^L \binom{L}{\ell_1} \gamma^{\ell_1} (1-\gamma)^{L-\ell_1} \frac{M/2}{M-1} P_s(e|\lambda_1), \quad (4.2-9a)$$

where

$$\gamma = \Pr\{\text{channel 1 jammed on hop } k\}. \quad (4.2-9b)$$

Noting that (4.2-8) and (4.2-9) are mathematically identical to equations (4-26) in [1], we observe that MFSK and RMFSK give equivalent PBNJ performances for individual channel AGC normalization.

4.3 ANALYSIS OF FH/RMFSK PERFORMANCE USING ANY-CHANNEL-JAMMED AGC SCHEME

In what follows we find the probability of bit error for the FH/RMFSK receiver using the any-channel-jammed AGC normalization scheme (ACJ).

4.3.1 Conditional Probability Of Error.

The probability of a symbol error, given the jamming event $(\underline{\lambda}_0, \underline{\lambda})$, is

$$P_s(e|\underline{\lambda}_0, \underline{\lambda}) = 1 - P_s(C|\underline{\lambda}_0, \underline{\lambda})$$

$$= 1 - \int_0^\infty d\alpha p_{z_1}(\alpha; \underline{\lambda}_0, \underline{\lambda}_1) \prod_{m=2}^M \int_0^\alpha d\beta_m p_{z_m}(\beta_m; \underline{\lambda}_0, \underline{\lambda}_m). \quad (4.3-1)$$

$$\begin{aligned}
\text{Since } P_s(C|\ell_0, \underline{\ell}) &= \Pr\{z_2 < z_1, z_3 < z_1, \dots, z_M < z_1\} \\
&= \Pr\{Kz_2 < Kz_1, Kz_3 < Kz_1, \dots, Kz_M < Kz_1\}, \tag{4.3-2}
\end{aligned}$$

we may analyze the error probability using the statistics $\{u_m\}$ instead of $\{z_m\}$
where $u_1 \equiv Kz_1 = x^2[2(\ell_0 - \ell_1); 2(\ell_0 - \ell_1)\rho_N]$

$$+ Kx^2[2(L + \ell_1 - \ell_0); 2(L - \ell_0)\rho_N + 2\ell_1\rho_T] \tag{4.3-3a}$$

and

$$u_m \equiv Kz_m = x^2[2(\ell_0 - \ell_m)] + Kx^2[2(L + \ell_m - \ell_0)] \quad m > 1. \tag{4.3-3b}$$

From Appendix A, the pdf's of these random variables are as follows: for the signal channel,

$$p_{u_1}(\alpha) = \begin{cases} p_{\chi^2}(\alpha; 2L, 2L\rho_N), & \ell_0 - \ell_1 = L \\ \frac{1}{K} p_{\chi^2}[\alpha/K; 2L, 2(L - \ell_0)\rho_N + 2\ell_1\rho_T], & \ell_0 - \ell_1 = 0; \\ \sum_{n=0}^{\infty} c_n p_{\chi^2}[\alpha; 2L + 2n, 2(\ell_0 - \ell_1)\rho_N], & 0 < \ell_0 - \ell_1 < L; \end{cases} \tag{4.3-4a}$$

where

$$c_n = e^{-(L - \ell_0)\rho_N - \ell_1\rho_T} \left(\frac{K-1}{K}\right)^n \left(\frac{1}{K}\right)^{L - \ell_0 + \ell_1} \ell_n^{L - \ell_0 + \ell_1 - 1} \frac{[-(L - \ell_0)\rho_N - \ell_1\rho_T]}{K-1}. \tag{4.3-4d}$$

For the nonsignal channels ($m > 1$),

$$p_{u_m}(\alpha) = \begin{cases} p_{\chi^2}(\alpha; 2L), & \ell_0 - \ell_m = L \\ \frac{1}{K} p_{\chi^2}(\alpha/K; 2L), & \ell_0 - \ell_m = 0 \end{cases} \quad (4.3-5a)$$

$$p_{u_m}(\alpha) = \frac{1}{K} p_{\chi^2}(\alpha/K; 2L), \quad \ell_0 - \ell_m = 0 \quad (4.3-5b)$$

$$\sum_{n=0}^{\infty} b_n p_{\chi^2}(\alpha; 2L+2n), \quad 0 < \ell_0 - \ell_m < L \quad (4.3-5c)$$

using the coefficients

$$b_n = \left(\frac{K-1}{K}\right)^n \left(\frac{1}{K}\right)^{L-\ell_0+\ell_m} \binom{n+L-\ell_0+\ell_m-1}{n} \quad (4.3-5d)$$

The symbol error probability expression for the $\{u_m\}$ statistics is (4.3-i) with the subscripts $\{u_m\}$ instead of $\{z_m\}$, $m=1, 2, \dots, M$.

4.3.1.1 Formulations of nonsignal channel probabilities

Since the nonsignal channel pdf's are identical except for the parameters $\{\ell_m\}$, the number of hops jammed in the individual channels, we may express the product

$$\prod_{m=2}^M \int_0^\alpha d\beta_m p_{u_m}(\beta_m; \ell_0, \ell_m) \prod_{m=2}^M \Pr\{u_m < \alpha\} \quad (4.3-6)$$

in terms of the numbers of channels with certain combinations of ℓ_0 and the $\{\ell_m\}$. The probabilities needed are

$$\Pr\{u_m < \alpha\} = \begin{cases} 1 - \Gamma(L; \alpha/2)/\Gamma(L), & \ell_0 - \ell_m = L \end{cases} \quad (4.3-7a)$$

$$\Pr\{u_m < \alpha\} = \begin{cases} 1 - \Gamma(L; \alpha/2K)/\Gamma(L), & \ell_0 - \ell_m = 0 \end{cases} \quad (4.3-7b)$$

$$\Pr\{u_m < \alpha\} = \begin{cases} 1 - \sum_{n=0}^{\infty} b_n \Gamma(L+n; \alpha/2)/\Gamma(L+n), & 0 < \ell_0 - \ell_m < L. \end{cases} \quad (4.3-7c)$$

J. S. LEE ASSOCIATES, INC.

Upon comparing (4.3-7) with (3.1-11), we observe that

$$\Pr\{u_m < \alpha\} \equiv F_L(\alpha; L - \ell_0 + \ell_m). \quad (4.3-8)$$

Thus we can write the product in (4.3-6) as

$$\prod_{m=2}^M \Pr\{u_m < \alpha\} = [F_L(\alpha; 0)]^{n_0} [F_L(\alpha; 1)]^{n_1} \dots [F_L(\alpha; L)]^{n_L} \quad (4.3-9)$$

where

$$n_i \triangleq \# (\text{channels with } L - \ell_0 + \ell_m = i), \quad (4.3-10)$$

and $[F_L(\alpha; p)]^{n_p}$ is given by (3.1-14), (3.1-17), and (3.1-28).

4.3.1.2 Formulation of symbol error probability in terms of previous results (Section 3).

If we denote the conditional probability of symbol error for the square-law linear combining FH/RMFSK receiver studied in Section 3 by

$$P_s(e; \rho_N, \rho_T | \underline{\ell})_{LC}, \quad (4.3-11)$$

we can by analogy express the conditional probability of symbol error for the any-channel-jammed AGC receiver as

$$P_s(e; \rho_N, \rho_T | \underline{\ell}_0, \underline{\ell})_{ACJ} = P_s(e; \rho_N, \rho'_T | \underline{v})_{LC} \quad (4.3-12a)$$

where

$$\rho'_T = \begin{cases} \frac{\ell_1 \rho_T + (L - \ell_0) \rho_N}{L - \ell_0 + \ell_1}, & \ell_0 - \ell_1 \neq L \\ \rho_T, & \ell_0 - \ell_1 \end{cases} \quad (4.3-12b)$$

J. S. LEE ASSOCIATES, INC.

and

$$\underline{v} \triangleq (L - \ell_0 + \ell_1, L - \ell_0 + \ell_2, \dots, L - \ell_0 + \ell_M). \quad (4.3-13)$$

4.3.2 Enumeration and Probabilities for Jamming Events.

The enumeration of jamming events, and their probabilities, has already been accomplished in Section 2.2 for the situation in which the jamming event is sufficiently described by the vector $\underline{\ell} = (\ell_1, \ell_2, \dots, \ell_M)$. Now our task is to develop for use in (4.3-12) the conditional probabilities $\Pr\{\ell_0 | \underline{\ell}\}$ for the parameter ℓ_0 , the number of hops on which at least one channel is jammed, given the vector $\underline{\ell}$.

The enumeration technique treated in Section 2.2 recognized the arbitrariness of the channel numbers m for $m > 1$ (nonsignal channels) by assuming that the calculations will generate the partially ordered $\underline{\ell}$ vector

$$\underline{\ell}' = \{(\ell_1, \ell_2, \dots, \ell_M) : \ell_2 \leq \ell_3 \leq \dots \leq \ell_M\}. \quad (4.3-14)$$

Thus, with this ordering the range of ℓ_0 is

$$\ell_X \triangleq \max(\ell_1, \dots, \ell_M) \leq \ell_0 \leq \min(L, \ell_1 + \ell_2 + \dots + \ell_M). \quad (4.3-15)$$

The number of elementary or $[v]$ -matrix jamming events characterized by a given $\underline{\ell}$ or $\underline{\ell}'$ vector is

$$\#([v] \rightarrow \underline{\ell}) = \binom{L}{\ell_1} \binom{L}{\ell_2} \dots \binom{L}{\ell_M} = \prod_{m=1}^M \binom{L}{\ell_m}, \quad (4.3-16)$$

J. S. LEE ASSOCIATES, INC.

and each $\underline{\ell}$ vector represents

$$\binom{M-1}{n_0, n_1, \dots, n_L}, \quad n_k = \#\{\ell_m = k\}, \quad m > 1 \quad (4.3-17)$$

$\underline{\ell}$ vectors. Thus

$$\sum_{\underline{\ell}} \binom{M-1}{n_0, n_1, \dots, n_L} \Pr\{\underline{\ell}; M, L\} = 1. \quad (4.3-18)$$

Now for jamming events specified by ℓ_0 as well as $\underline{\ell}$, the number of elementary jamming events thus specified can be shown to be

$$\#([\nu] \rightarrow \ell_0, \underline{\ell}) = \binom{L}{\ell_0} \sum_{r=0}^{\ell_0 - \ell_X} \binom{\ell_0}{r} (-1)^r \prod_{m=1}^M \binom{\ell_0 - r}{\ell_m}. \quad (4.3-19)$$

For example, if $\ell_0 = \ell_X = \max(\ell_m)$, there are

$$\binom{L}{\ell_0} \prod_{m=1}^M \binom{\ell_0}{\ell_m} \quad (4.3-20)$$

$[\nu]$ events. Summation of (4.3-19) over the values of ℓ_0 given by (4.3-15) can be shown numerically to give (4.3-16). (See Appendix B.4.)

What is needed for evaluation of the ACJ total probability of error are the probabilities of the jamming events $(\ell_0, \underline{\ell})$ and the number of $(\ell_0, \underline{\ell})$ events represented by the ordered version. Since $(\ell_0, \underline{\ell})$ is a subset of $\underline{\ell}$ for any $\underline{\ell}$, and permutations of the nonsignal channel elements of $\underline{\ell}$ do not affect ℓ_0 , it is reasonable that (4.3-17) gives the required number. This fact is confirmed

J. S. LEE ASSOCIATES, INC.

by the consideration that

$$\Pr\{\underline{\ell}'; M, L\} = \sum_{\ell_0} \Pr\{\underline{\ell}', \ell_0; M, L\} \quad (4.3-21)$$

which can be substituted in (4.3-18) to show that $\Pr\{\ell_0, \underline{\ell}\}$ must be multiplied by (4.3-17).

The probability of the event $(\ell_0, \underline{\ell})$ is derived in the following manner:

$$\begin{aligned} \Pr\{\underline{\ell}, \ell_0\} &= \binom{L}{\ell_0} \Pr\{\underline{v}_1 + \underline{v}_2 + \dots + \underline{v}_{\ell_0} = \underline{\ell}, \underline{v}_1 \neq 0, \underline{v}_2 \neq 0, \dots, \\ &\quad \underline{v}_{\ell_0} \neq 0, \underline{v}_{\ell_0+1} = 0, \dots, \underline{v}_L = 0\} \\ &= \binom{L}{\ell_0} \pi_0^{L-\ell} \Pr\{\underline{v}_1 + \underline{v}_2 + \dots + \underline{v}_{\ell_0} = \underline{\ell}, \underline{v}_1 \neq 0, \dots, \underline{v}_{\ell_0} \neq 0\} \end{aligned} \quad (4.3-22)$$

The probability required in (4.3-22) can be computed using the convolutional method described in Section 2.2.5, modified to give

$$\begin{aligned} \Pr\{\underline{v}_1 + \underline{v}_2 + \dots + \underline{v}_{\ell_0} = \underline{\ell}, \underline{v}_1 \neq 0, \underline{v}_2 \neq 0, \dots, \underline{v}_{\ell_0} \neq 0\} \\ = \sum_{\underline{v}_1 > 0} \sum_{\underline{v}_2 > 0} \dots \sum_{\underline{v}_{\ell_0} > 0} \Pr\{\underline{v}_1\} \Pr\{\underline{v}_2\} \dots \Pr\{\underline{v}_{\ell_0}\} \delta\left[\underline{\ell} - \sum_{r=1}^{\ell_0} \underline{v}_r\right]. \end{aligned} \quad (4.3-23)$$

This method is useful for M tending to be large; for $M=2$, it is simpler to recognize that (2.2-24), repeated here as

$$\begin{aligned} \Pr\{\underline{\ell}; 2, L\} &= \sum_{n=0}^L \binom{L}{n, L-\ell_2-n, L-\ell_1-n, \ell_1+\ell_2+n-L} \\ &\times \pi_0^n \pi_1^{2L-\ell_1-\ell_2-2n} \pi_2^{\ell_1+\ell_2-L+n}, \end{aligned} \quad (4.3-24)$$

J. S. LEE ASSOCIATES, INC.

is the sum of $\Pr\{\underline{e}, e_0; 2, L\}$ over e_0 , with $n \equiv L-e_0$. Therefore,

$$\Pr\{\underline{e}, e_0; 2, L\} = \binom{L}{L-e_0, e_0-e_2, e_0-e_1, e_1+e_2-e_0} \pi_1^{2e_0-e_1-e_2} \pi_2^{e_1+e_2-e_0}. \quad (4.3-25)$$

For M=4, we recognize the same principle in the equation given in Table 2.2-4
for $\Pr\{\underline{e}; 4, L\}$ to give the $\Pr\{e_0, \underline{e}; 4, L\}$ equation shown in Table 4.3-1.

TABLE 4.3-1 PROBABILITIES OF (v_0, \underline{v}) JAMMING EVENTS FOR $M = 4$

$$\Pr \{ \underline{v}_0, \underline{v}; 4, L \} = \frac{\sum_{n_1=0}^L \sum_{n_2=0}^L \dots \sum_{n_{15}=0}^L \frac{L!}{(L-n)! n_1! \dots n_{15}!}}{\pi_0^{L-n} \pi_1^{n_1} \pi_2^{n_2} \dots \pi_{15}^{n_{15}}} \times \pi_3^{n_7+n_{11}+n_{13}+n_{14}+n_4} \pi_4^{n_{15}}$$

CONSTRAINTS

CONSTRAINTS:

$$\begin{aligned}
 & n_1 + n_3 + n_5 + n_7 + n_9 + n_{11} + n_{13} + n_{15} = \lambda_1 \\
 & n_2 + n_3 + n_6 + n_7 + n_{10} + n_{11} + n_{14} + n_{15} = \lambda_2 \\
 & n_4 + n_5 + n_6 + n_7 + n_{12} + n_{13} + n_{14} + n_{15} = \lambda_3 \\
 & n_8 + n_9 + n_{10} + n_{11} + n_{12} + n_{13} + n_{14} + n_{15} = \lambda_4 \\
 & \sum_{i=1}^{15} n_i = \lambda_0
 \end{aligned}$$

J. S. LEE ASSOCIATES, INC.

4.4 NUMERICAL RESULTS

In this section we present numerical results for the performance of the individual-channel AGC receiver and the any channel-jammed (ACJ) AGC receiver.

4.4.1 Numerical Results for Individual-Channel AGC Receiver

The numerical computations for the performance of the individual-channel AGC receiver were performed using (4.2-8) and (4.2-9). In this particular case, no unusual computational difficulties are encountered in the computations. A listing of the computer program is given in Appendix E.

Figures 4.4-1 through 4.4-4 show the performance of binary ($M=2$) RMFSK/FH with $L=1, 2, 3$, and 4 hops/bit, respectively with the jamming fraction $\gamma=q/N$ as a parameter. We observe that the choice of jamming fraction is critical to the effective operation of the jammer. This is similar to the behavior of the square-law combining receiver. However, unlike the square-law combining receiver, the optimum jamming fraction against the individual-channel AGC receiver is $\gamma=1.0$ over a wider range of E_b/N_j , especially for higher values of L , the number of hops/bit.

Figure 4.4-5 compares the worst-case jamming performance of binary RMFSK/FH as L varies. We observe that over the range of about $E_b/N_j=8$ dB to $E_b/N_j=39$ dB, the optimum choice for the communicator is $L=2$ or 3 hops/bit. However, outside this range $L=1$ is optimum. In no case does increasing L beyond 3 hops/bit improve the performance.

Figures 4.4-6 through 4.4-9 show the performance of RMFSK/FH when $M=4$ and $L=1, 2, 3$, and 4, respectively. Again, the importance to the jammer of

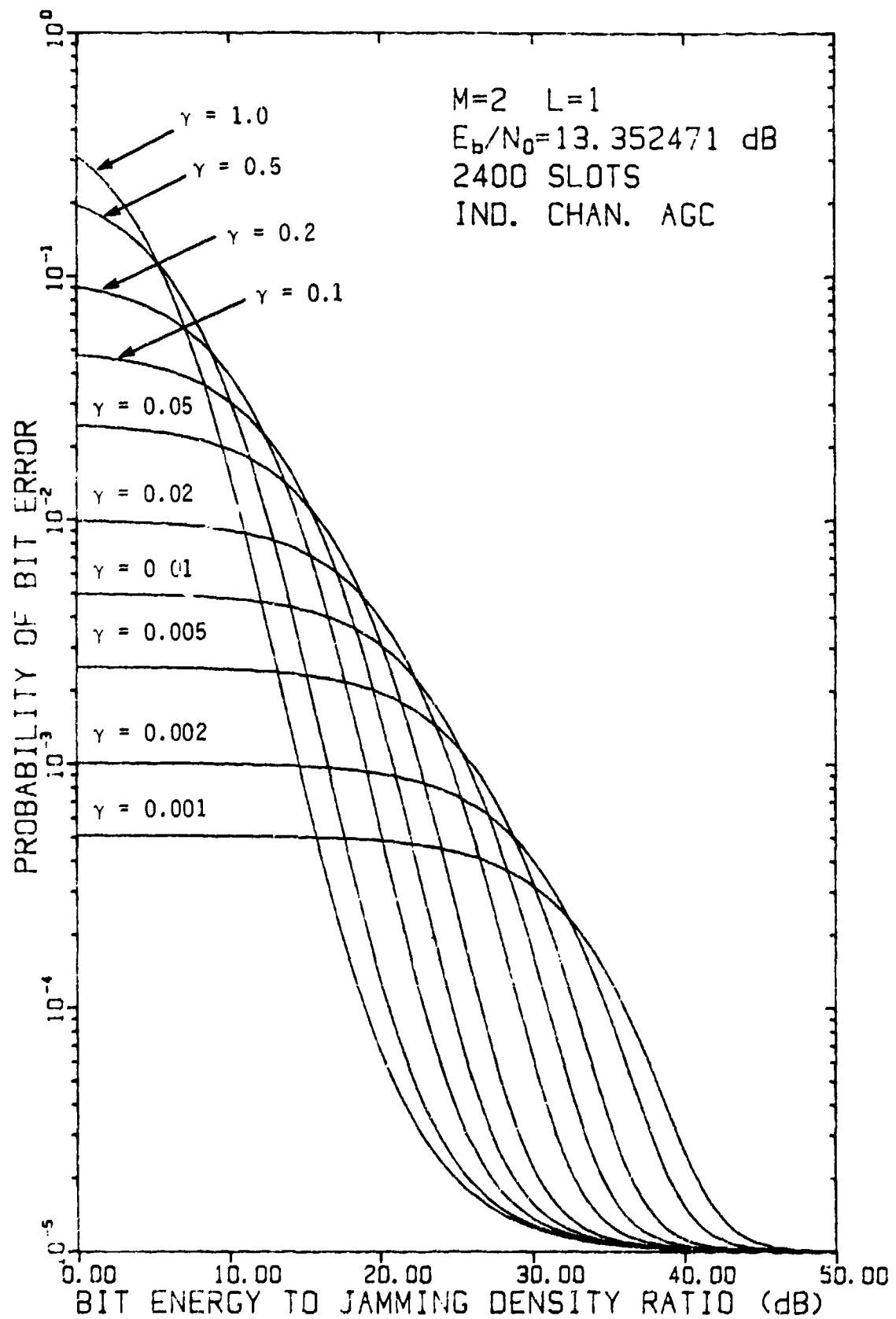


FIGURE 4.4-1 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH $M = 2$ AND $L = 1$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.352471 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L = 1$) WITH JAMMING FRACTION γ AS A PARAMETER

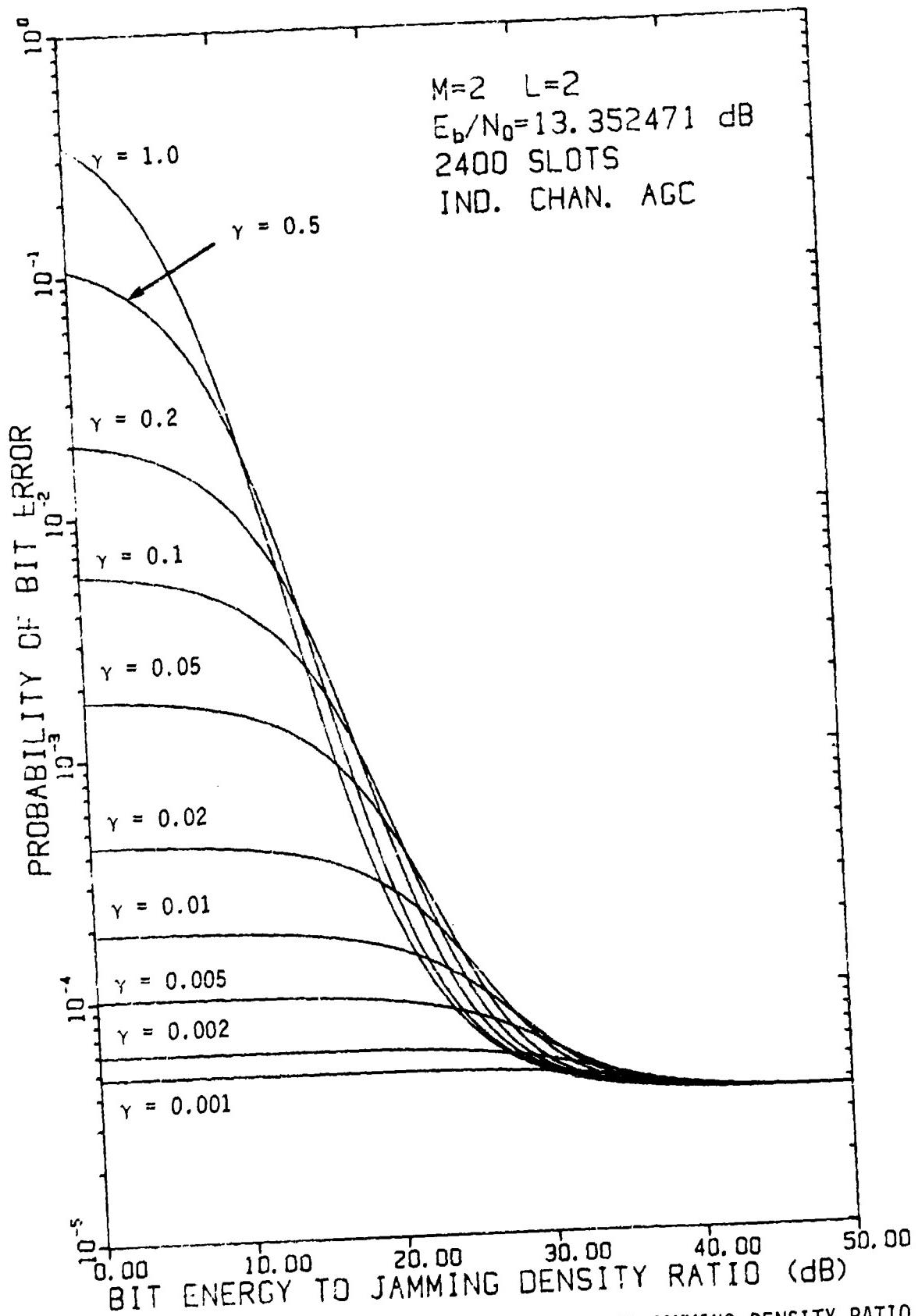


FIGURE 4.4-2 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH $M = 2$ AND $L = 2$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.352471$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L = 1$) WITH JAMMING FRACTION γ AS A PARAMETER

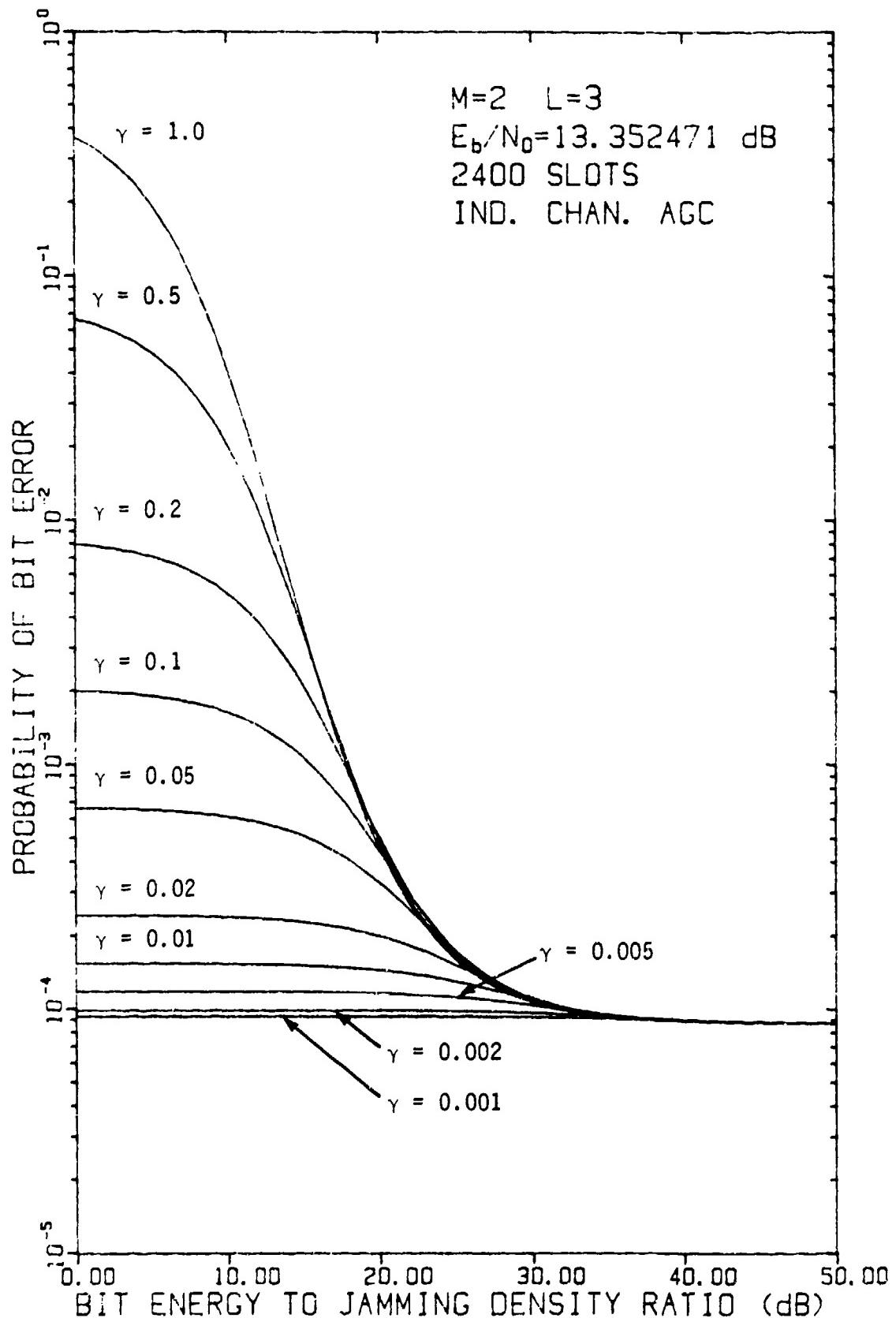


FIGURE 4.4-3 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH $M = 2$ AND $L = 3$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.352471 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L = 1$) WITH JAMMING FRACTION γ AS A PARAMETER

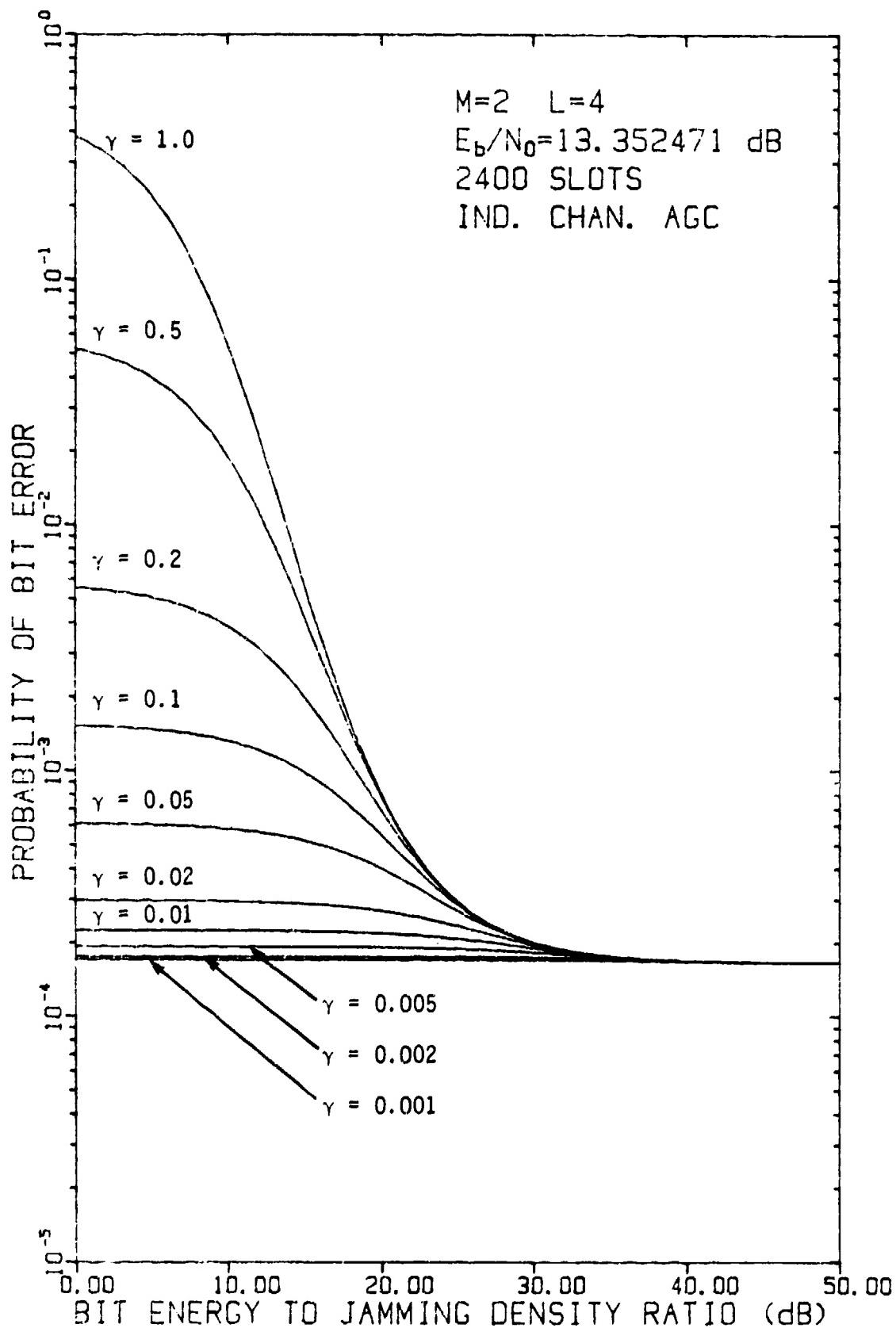


FIGURE 4.4-4 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH $M = 2$ AND $L = 4$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.352471$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L = 1$) WITH JAMMING FRACTION γ AS A PARAMETER

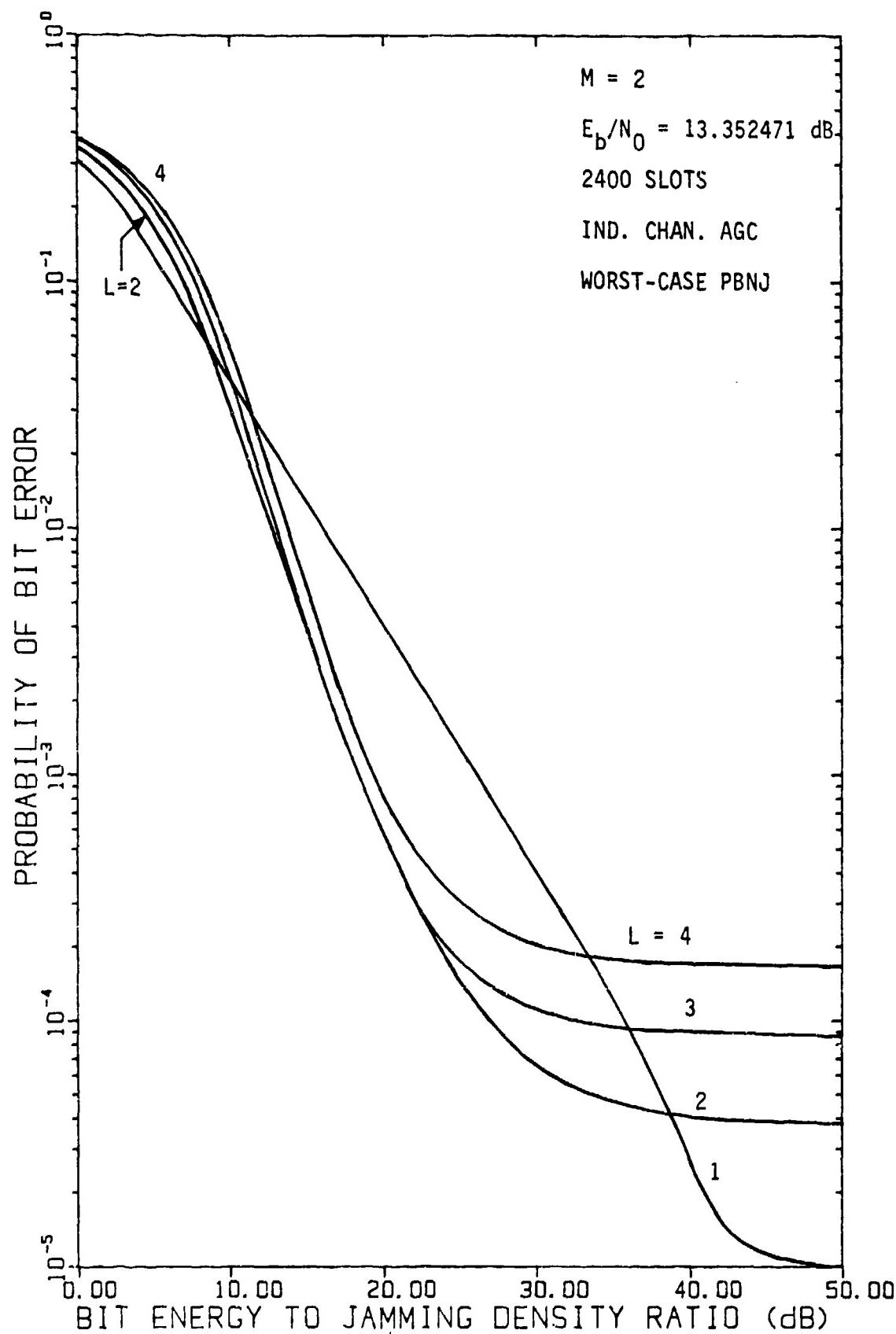


FIGURE 4.4-5 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER AND $M = 2$ WITH NUMBER OF HOPS/SYMBOL L AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING

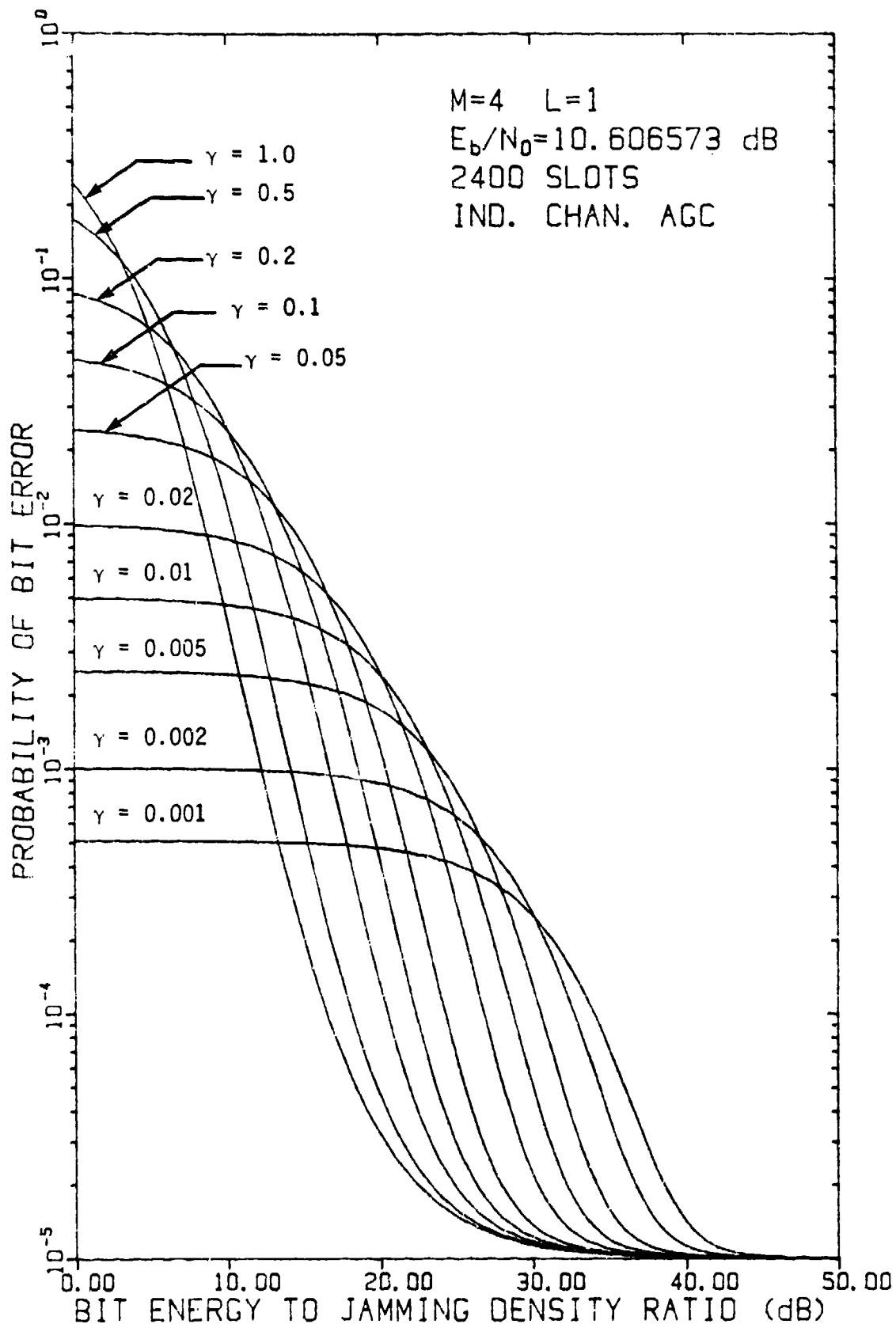


FIGURE 4.4-6 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH $M = 4$ AND $L = 2$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L = 1$) WITH JAMMING FRACTION γ AS A PARAMETER

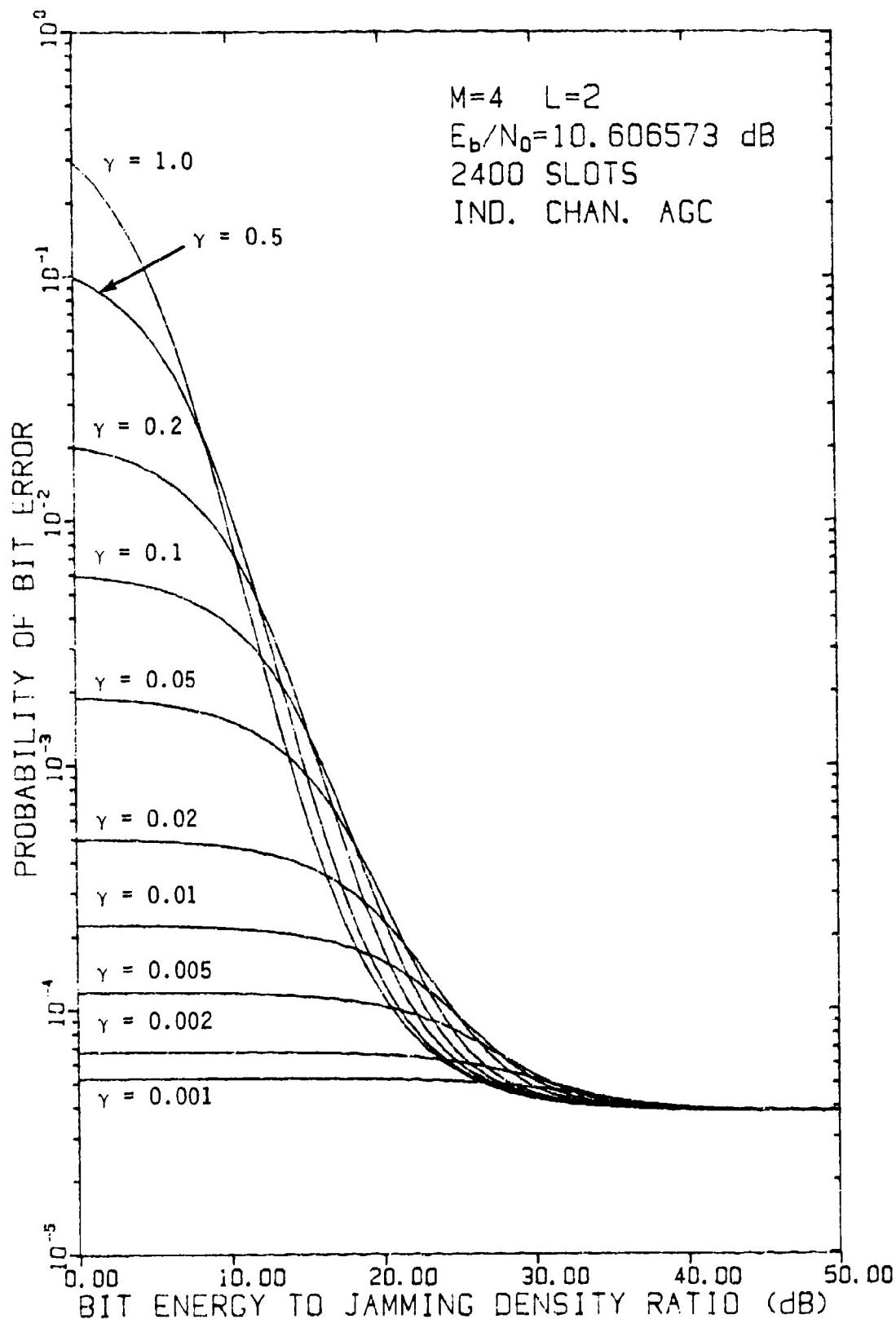


FIGURE 4.4-7 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH $M = 4$ AND $L = 2$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L = 1$) WITH JAMMING FRACTION γ AS A PARAMETER

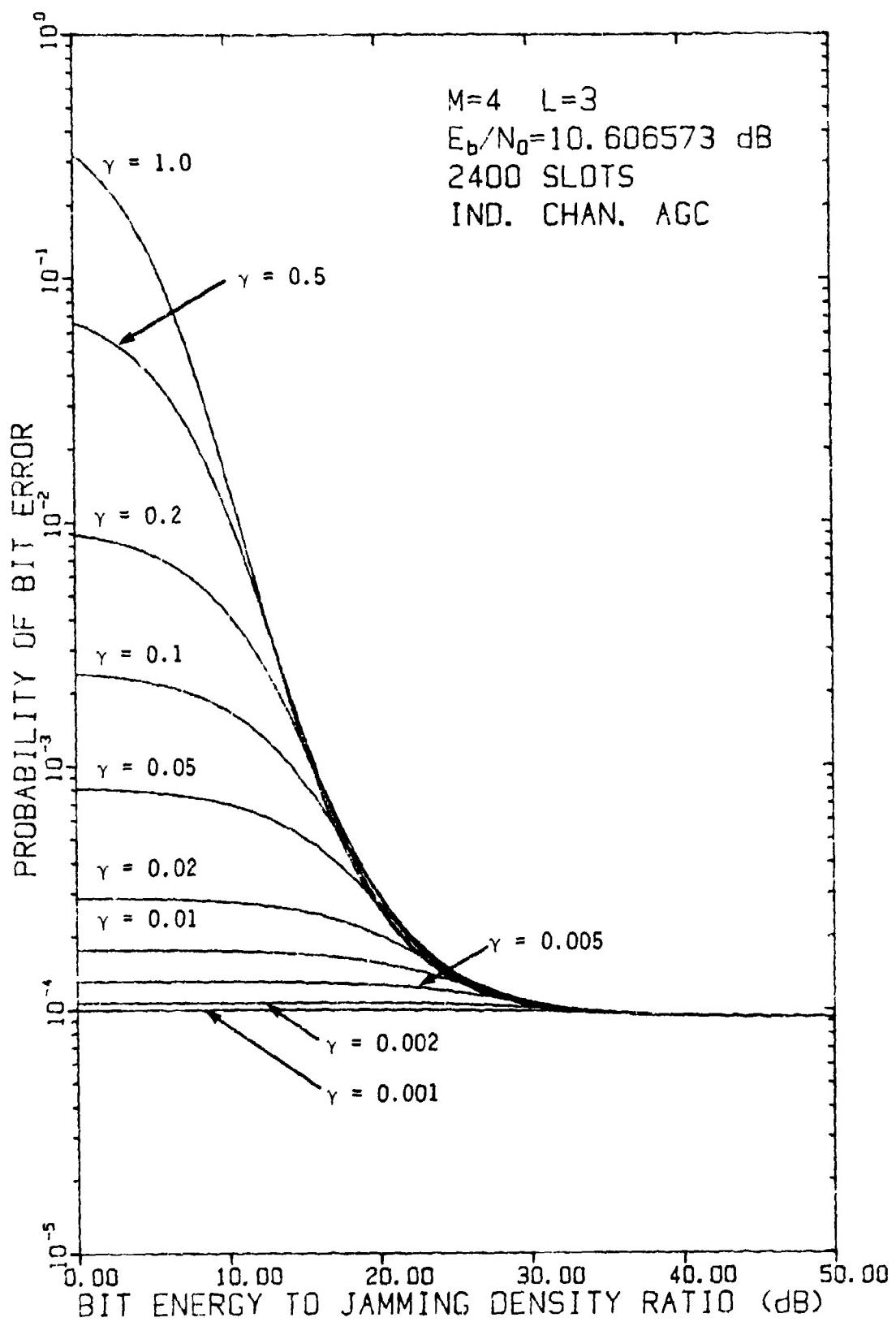


FIGURE 4.4-8 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH $M = 4$ AND $L = 3$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L = 1$) WITH JAMMING FRACTION γ AS A PARAMETER

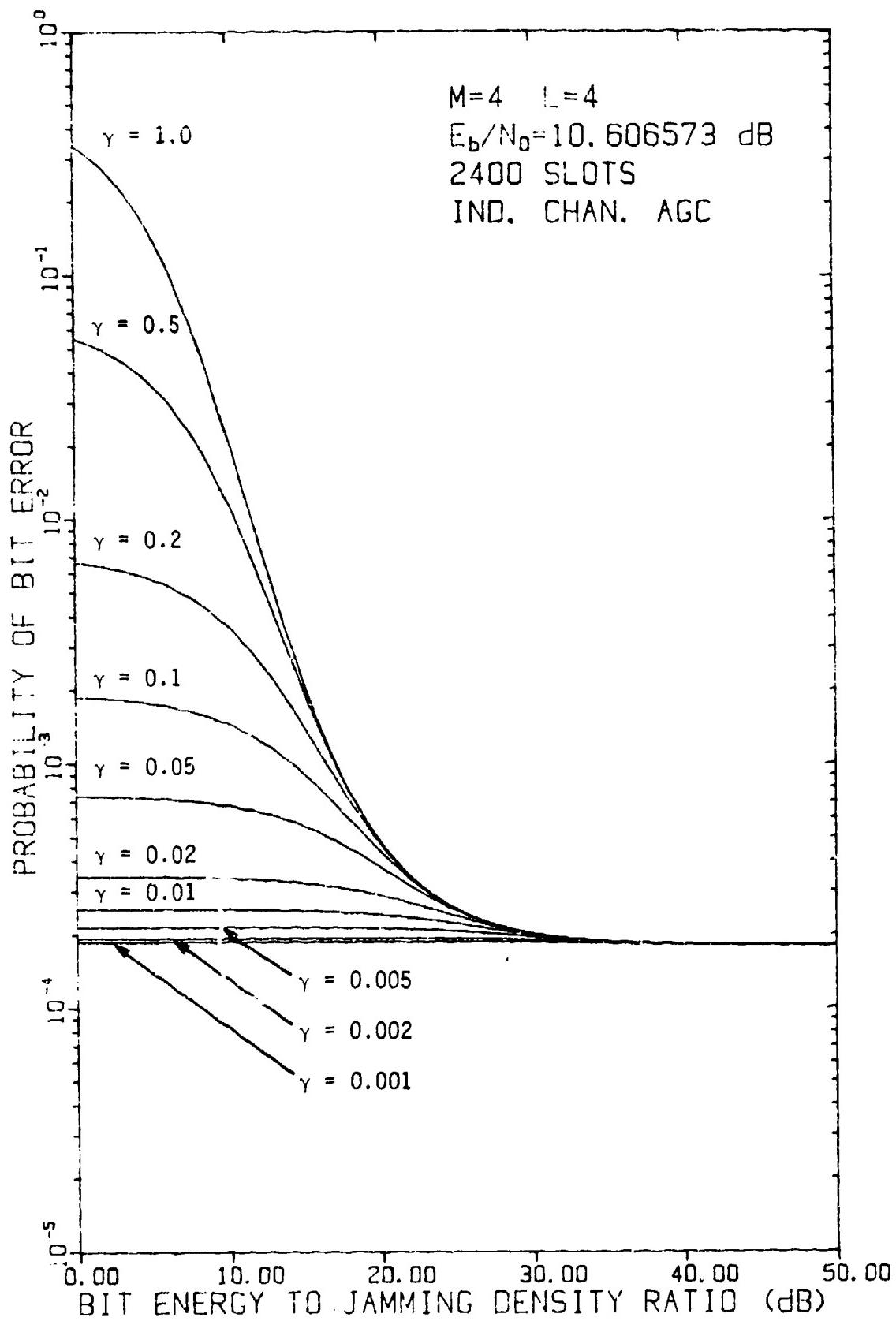


FIGURE 4.4-9 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH $M = 4$ AND $L = 4$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L = 1$) WITH JAMMING FRACTION γ AS A PARAMETER

the correct selection of γ stands out clearly. Figure 4.4-10 compares the performance of 4-ary RMFSK/FH in worst-case partial-band noise jamming as L varies. We see that for E_b/N_j in the range of about 7 to 36 dB, $L=2$ or 3 is optimum; elsewhere, $L=1$ is optimum.

Figures 4.4-11 through 4.4-14 show the curves for $M=8$ with γ as a parameter and $L=1, 2, 3$, and 4, respectively. Figure 4.4-15 shows performance for $M=8$ with L as a parameter in worst-case partial-band noise jamming. Again, from about $E_b/N_j=5$ dB to 35 dB the optimum L is 2 or 3, but elsewhere $L=1$ is optimum.

Two important conclusions can be drawn from these curves:

- The correct choice of fraction γ is critical for the jammer;
- The communicator can obtain only a small benefit by using multiple hops/symbol, and then only over a limited range of jamming conditions.

4.4.2 Numerical Results for Any-Channel-Jammed AGC (ACJ-AGC) Receiver

The numerical computations for the performance of the ACJ-AGC receiver required the use of two alternative forms. We used (4.3-12) for the computations, in conjunction with the computational techniques discussed in Section 3.3. The switch-over criteria for choosing between series and numerical integration were determined empirically. A listing of the computer program for numerical computations is given in Appendix F and a listing of the program which produced the plots is given in Appendix G. This latter program is typical of the plotting programs for all receivers; for sake of brevity only this one version of the plot program is included in our report.

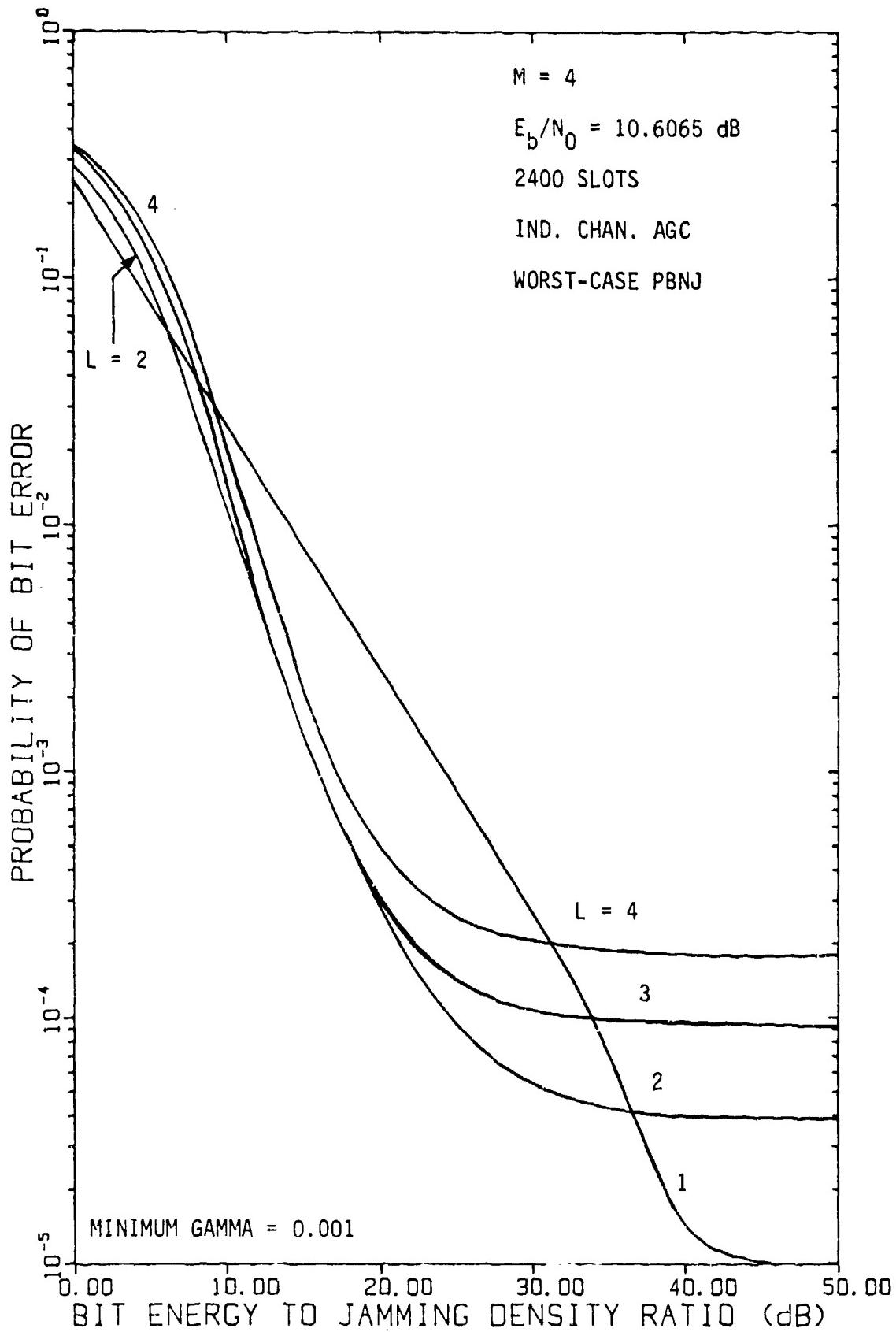


FIGURE 4.4-10 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER AND M = 4 WITH NUMBER OF HOPS/SYMBOL L AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING

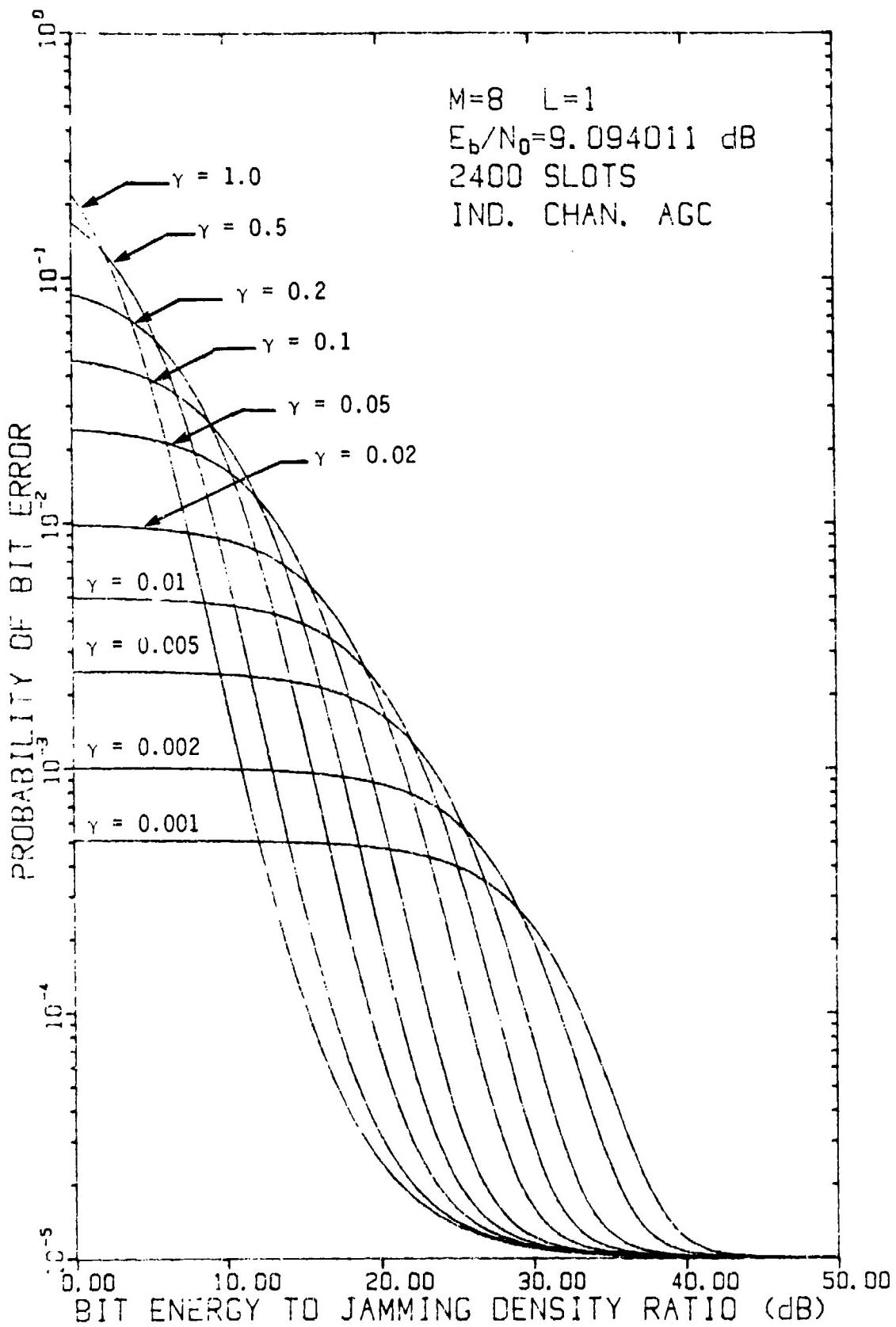


FIGURE 4.4-11 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH $M = 8$ AND $L = 1$ HOP/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 9.094011 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L = 1$) WITH JAMMING FRACTION γ AS A PARAMETER

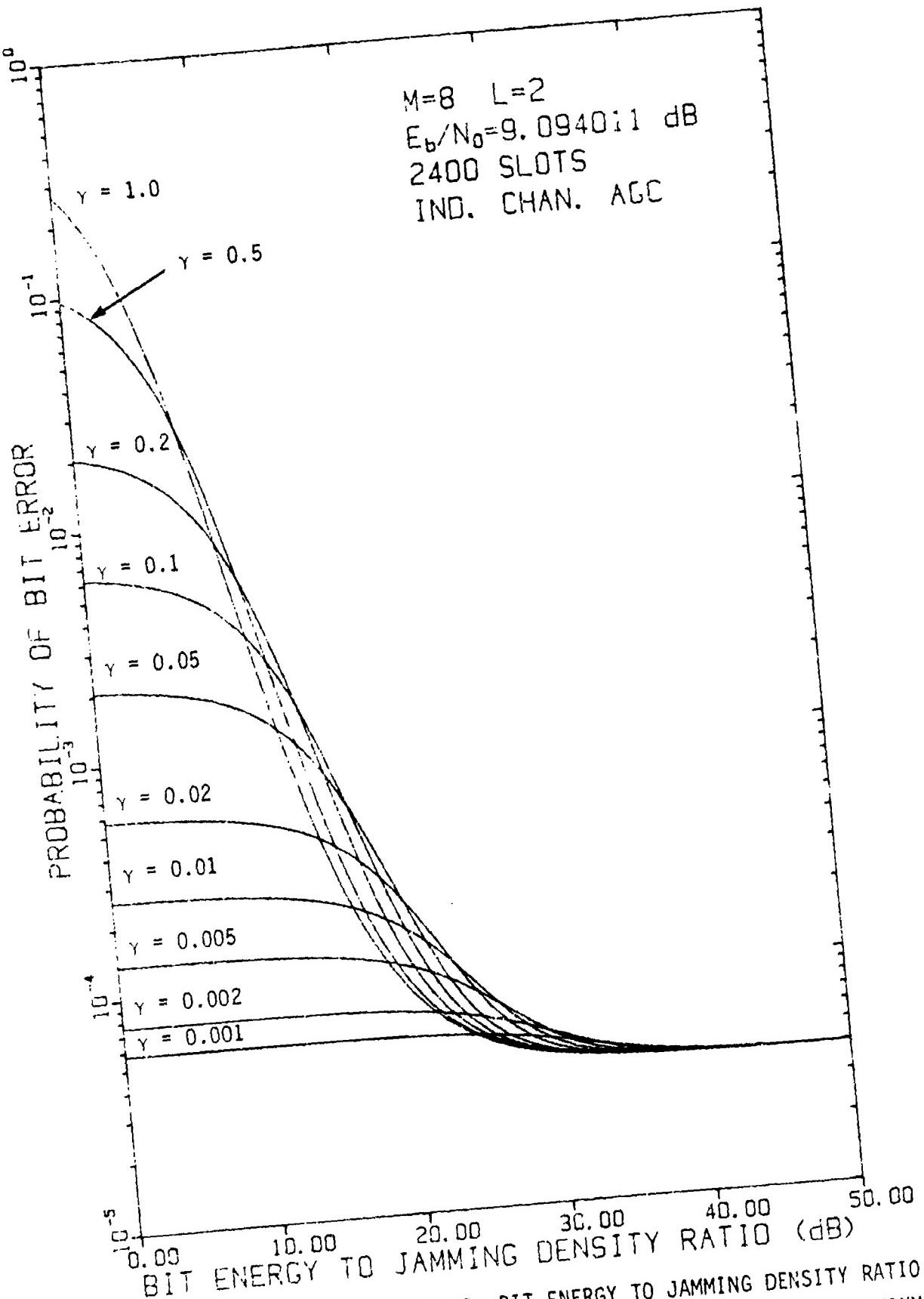


FIGURE 4.4-12 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH $M = 8$ AND $L = 2$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 9.094011$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L = 1$) WITH JAMMING FRACTION γ AS A PARAMETER

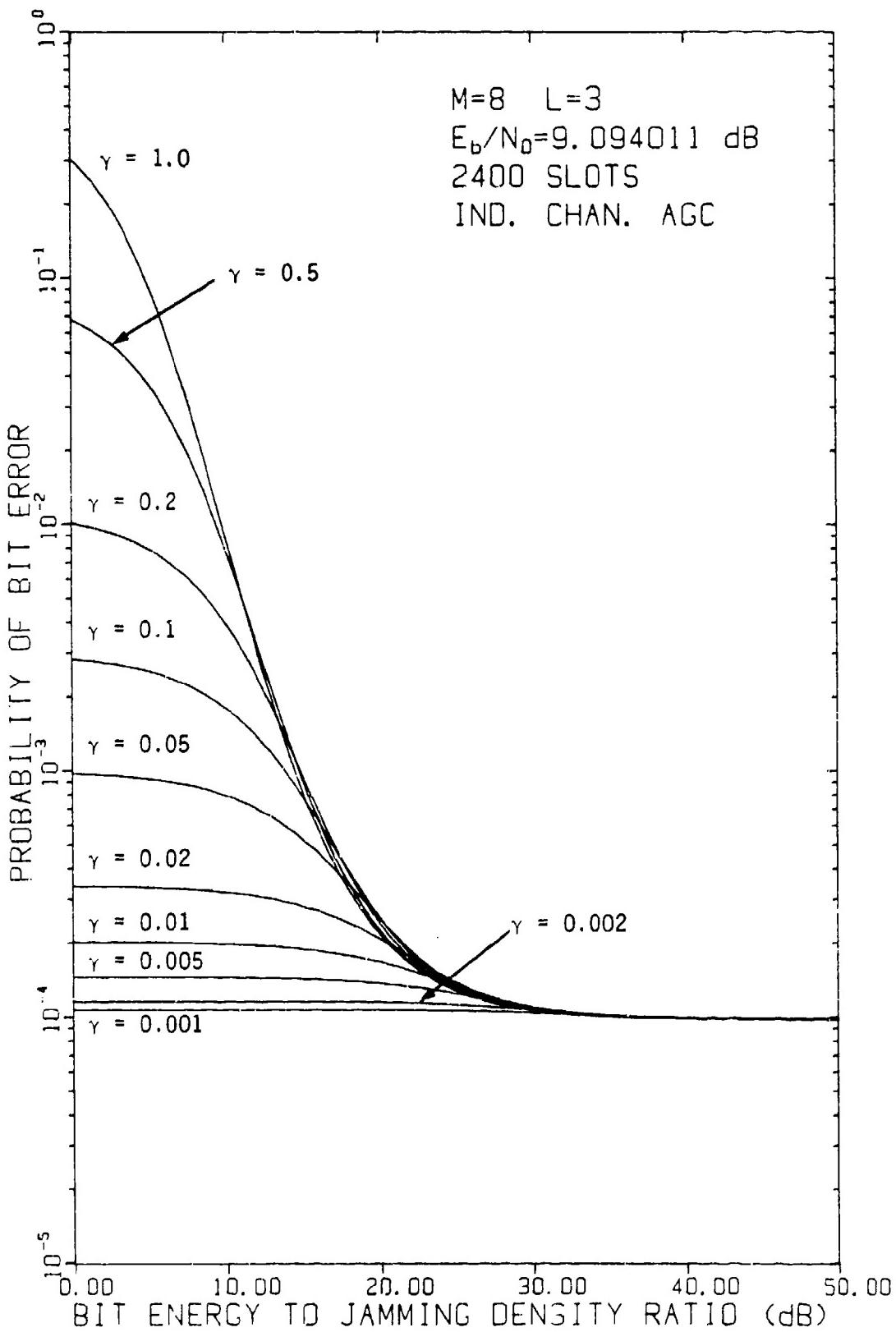


FIGURE 4.4-13 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH $M = 8$ AND $L = 3$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 9.094011$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L = 1$) WITH JAMMING FRACTION γ AS A PARAMETER

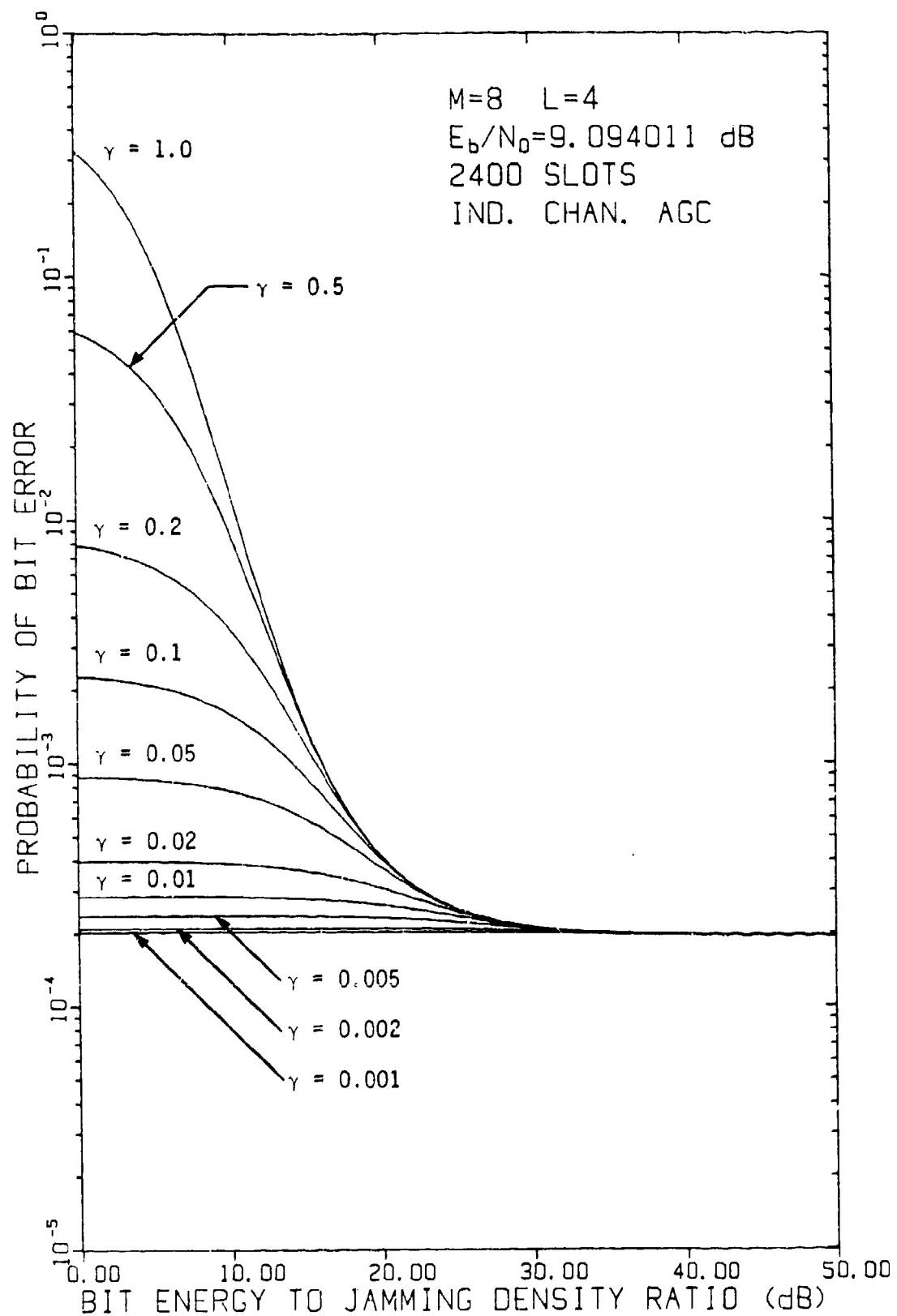


FIGURE 4.4-14 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER WITH $M = 8$ AND $L = 4$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 9.094011 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L = 1$) WITH JAMMING FRACTION γ AS A PARAMETER

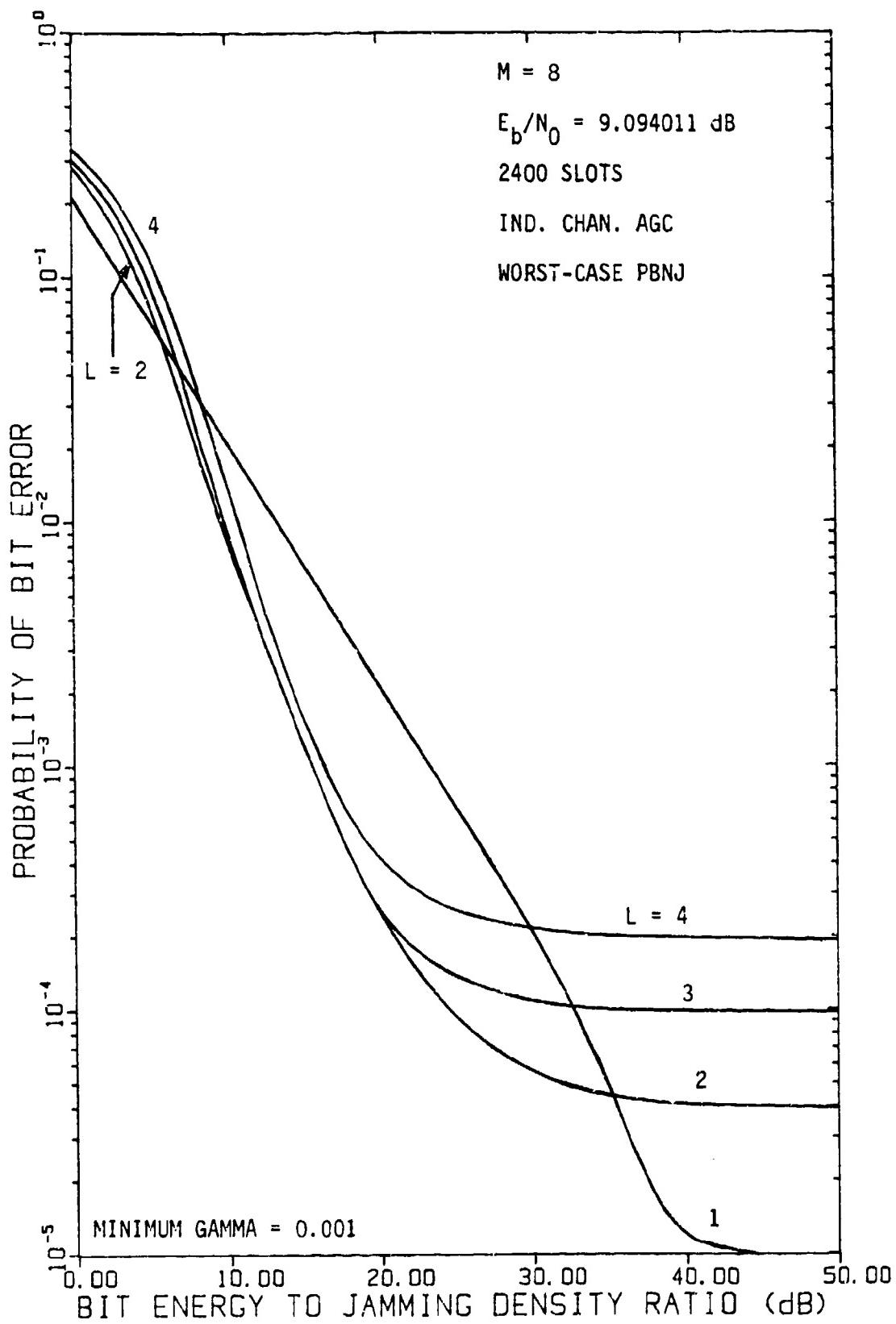


FIGURE 4.4-15 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER AND $M=8$ WITH NUMBER OF HOPS/SYMBOL AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING

J. S. LEE ASSOCIATES, INC.

Figures 4.4-16 through 4.4-19 show the performance of the ACJ-AGC receiver with $M=2$ for $L=1, 2, 3$, and 4 hops/bit, respectively. Figure 4.4-20 summarizes the performance in worst-case partial-band noise jamming with L as a parameter. Again, we find a limited range, roughly $10 \text{ dB} \leq E_b/N_j \leq 35 \text{ dB}$, where the optimum diversity is $L=2$ or 3 hops per bit; elsewhere $L=1$ is optimum.

Figures 4.4-21 through 4.4-24 show the performance of the ACJ-AGC receiver with $M=4$ for $L=1, 2, 3$, and 4 hops/symbol, respectively. These curves show the same general behavior as those for the other receivers. Figure 4.4-25 shows the performance in worst-case partial-band noise jamming with L as a parameter. Again, we find a limited range where $L=2$ or 3 is optimum, but elsewhere $L=1$ is optimum.

Finally, Figures 4.4-26 and 4.4-27 show the performance for $M=8$ and $L=1$ and 2, respectively. Because of the large computer time required, $L>2$ was not considered for $M=8$. Figure 4.4-28 summarizes performance in worst-case partial-band noise jamming. Again, there is a region where diversity ($L=2$) offers some advantage.

In summary, a small amount of diversity ($L=2$ or 3) is of meaningful benefit to RMFSK/FH over a limited range of signal-to-jamming ratios. However, outside this range no diversity ($L=1$) gives better performance. In some cases, e.g. Figure 4.4-20 for $M=2$, the penalty for using $L=3$ in the absence of jamming is nearly the same as the benefit of using $L=3$ when $E_b/N_j=25 \text{ dB}$.

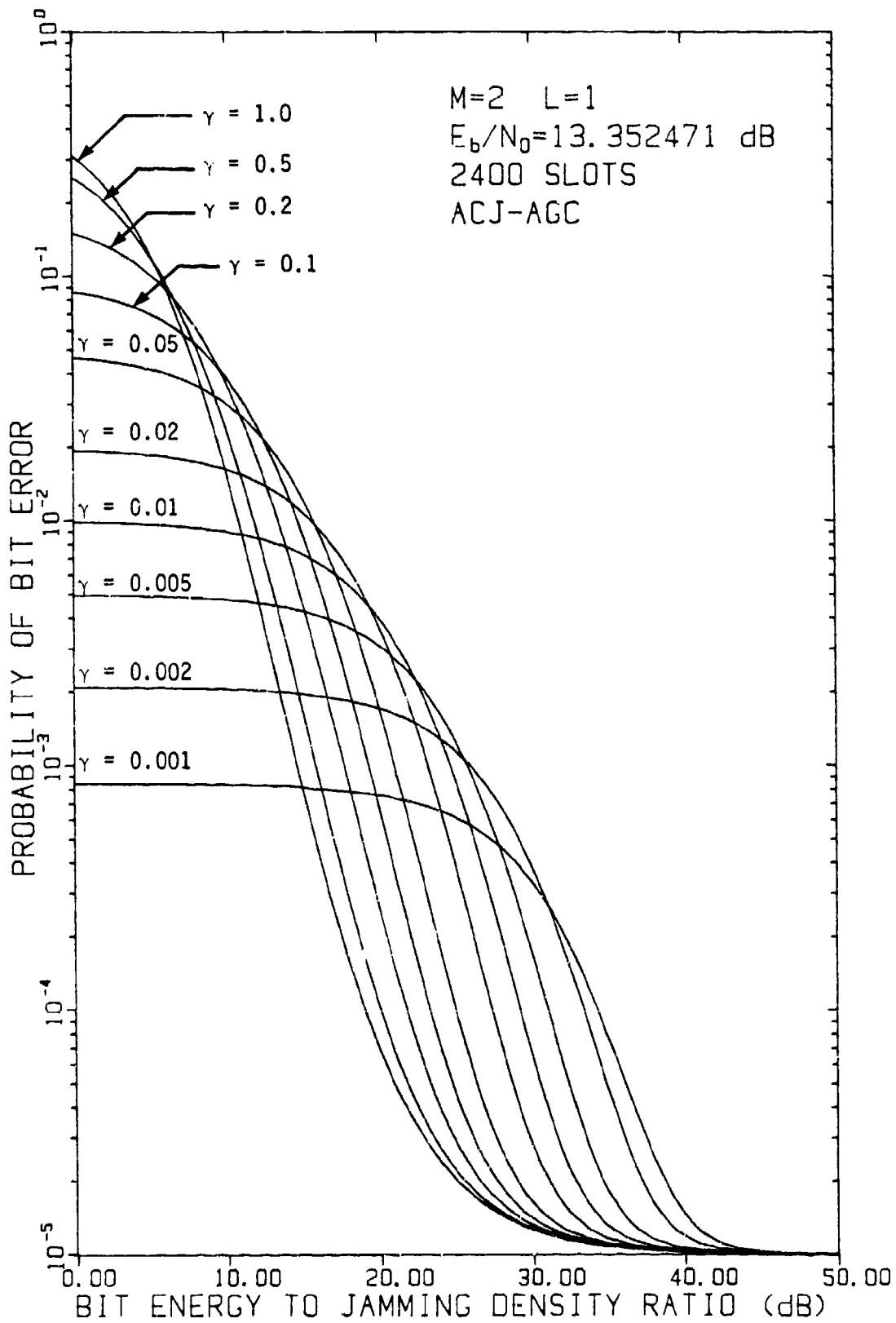


FIGURE 4.4-16 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER WITH $M=2$ AND $L=1$ HOP/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.352471$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L=1$) WITH JAMMING FRACTION γ AS A PARAMETER

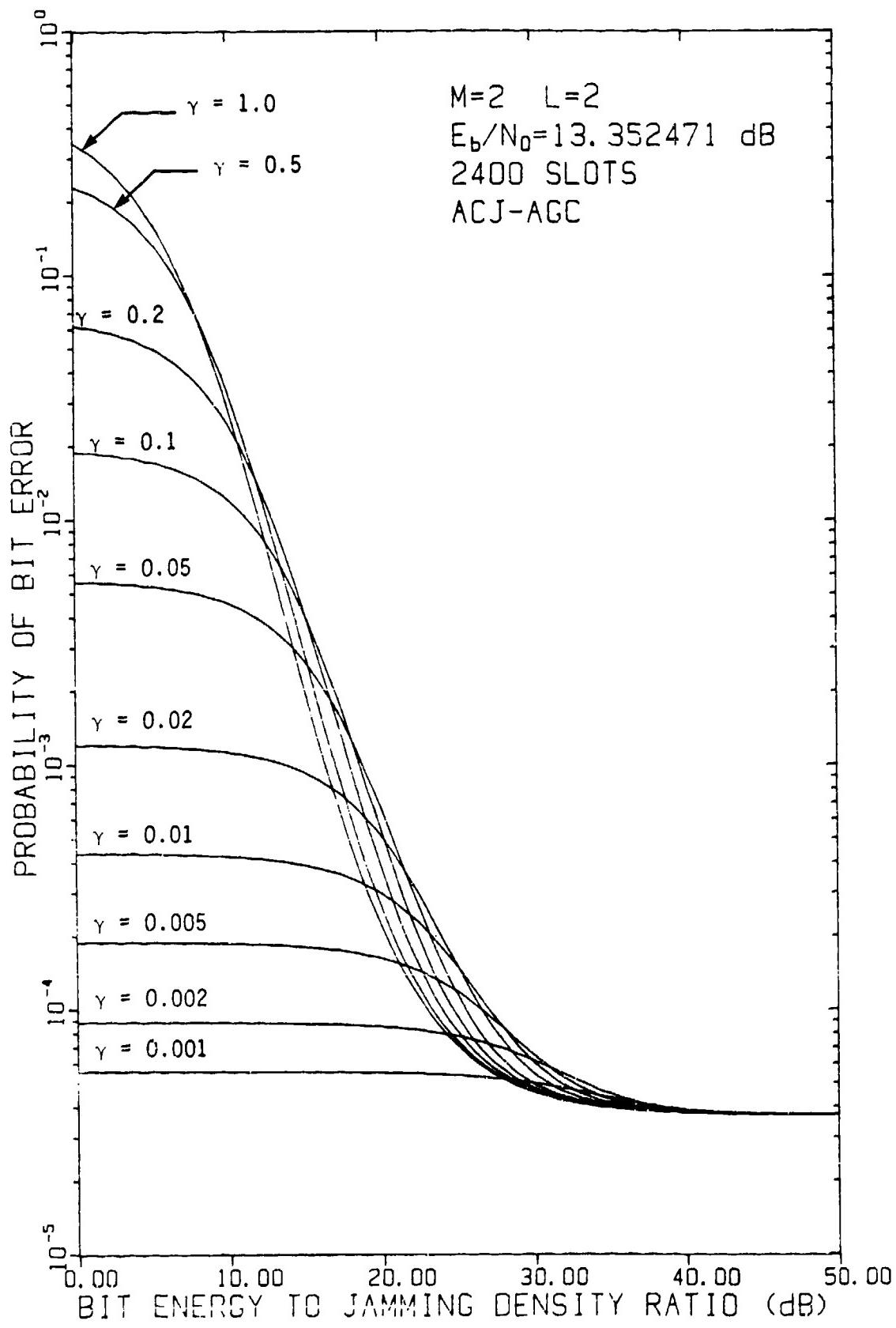


FIGURE 4.4-17 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER WITH $M=2$ AND $L=2$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.352471$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L=1$) WITH JAMMING FRACTION γ AS A PARAMETER

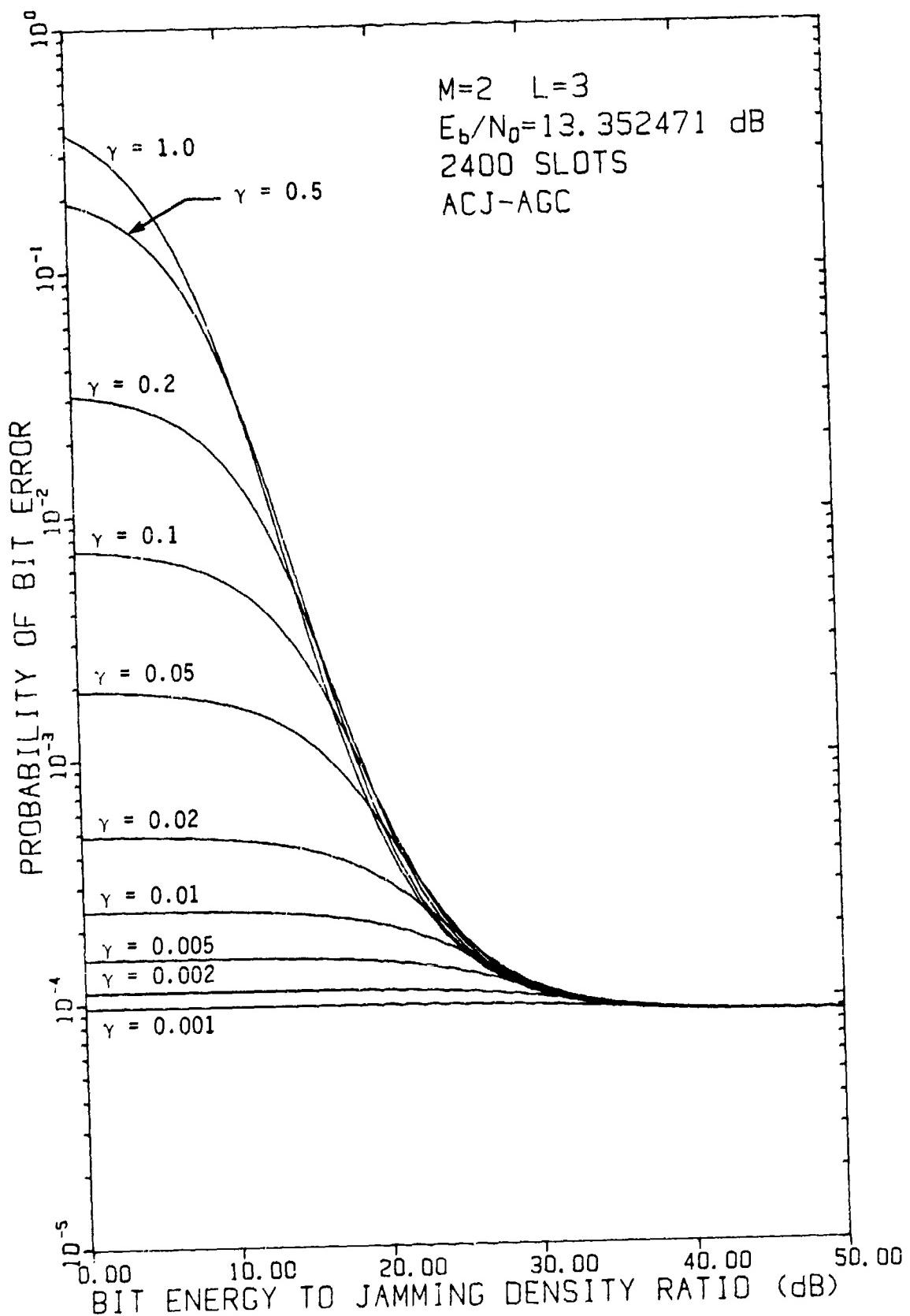


FIGURE 4.4-18 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER WITH $M=2$ AND $L=3$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.352471 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L=1$) WITH JAMMING FRACTION γ AS A PARAMETER

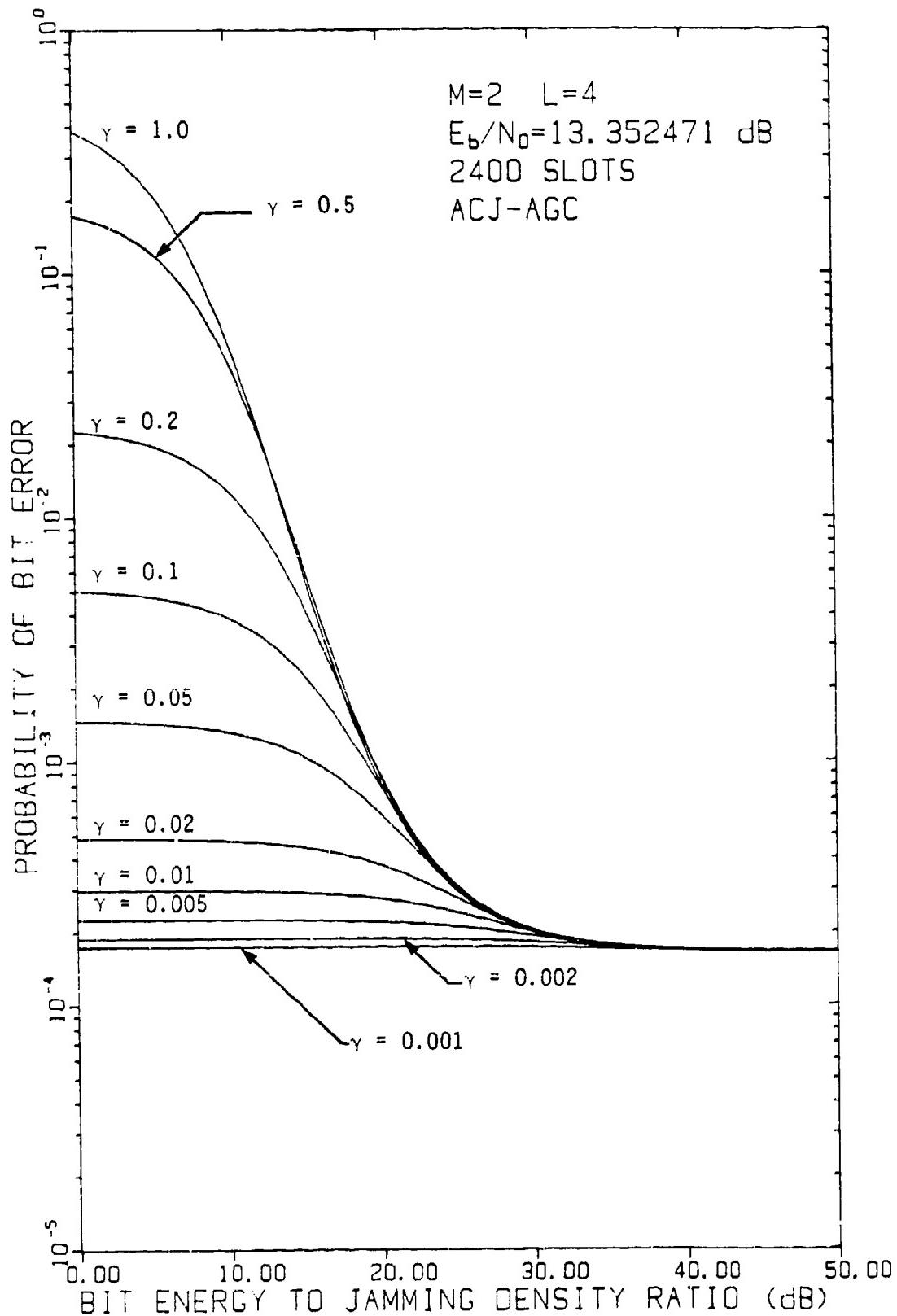


FIGURE 4.4-19 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER WITH M=2 AND L=4 HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.352471$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN L=1) WITH JAMMING FRACTION γ AS A PARAMETER

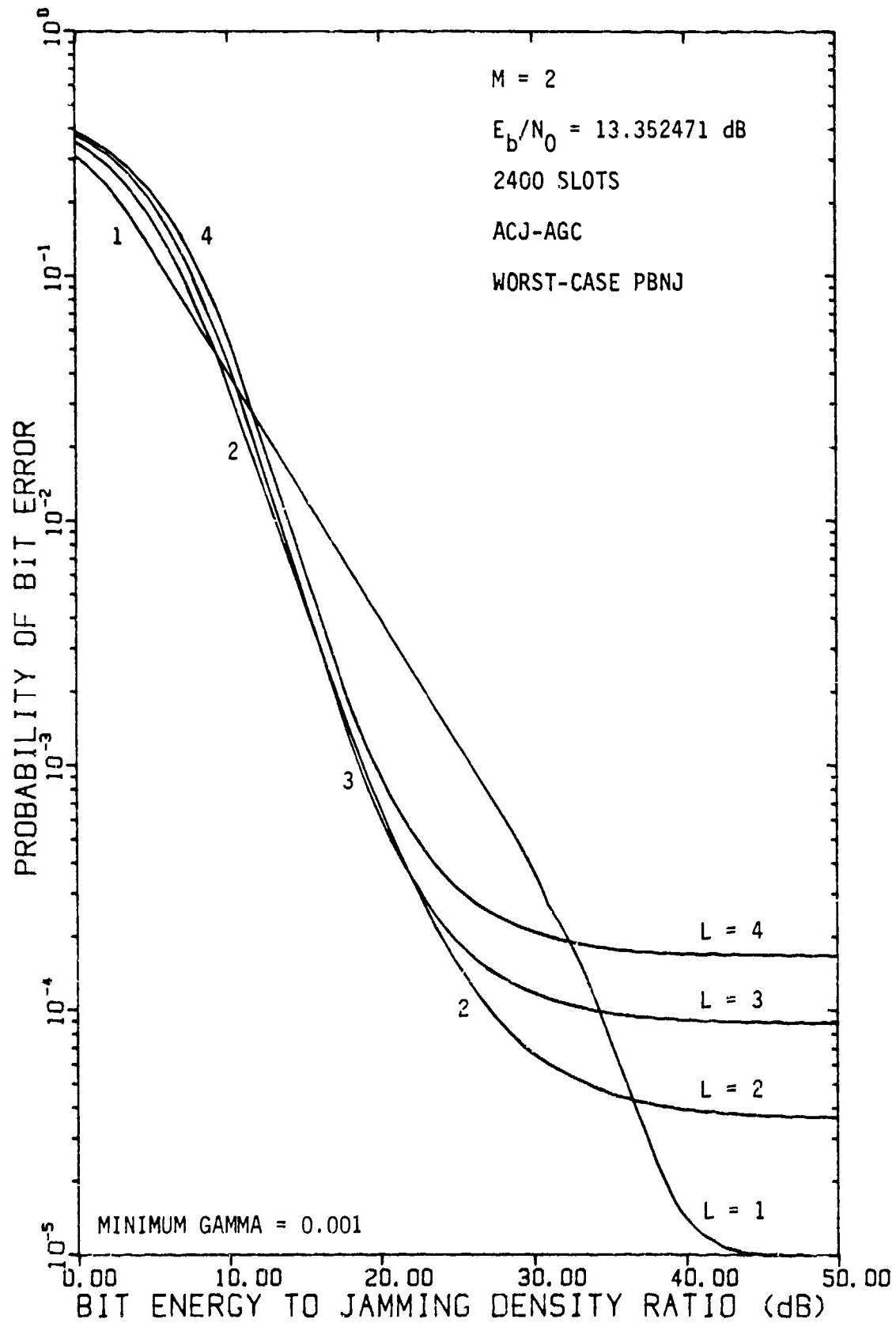


FIGURE 4.4-20 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER AND M=2 WITH NUMBER OF HOPS/BIT AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING

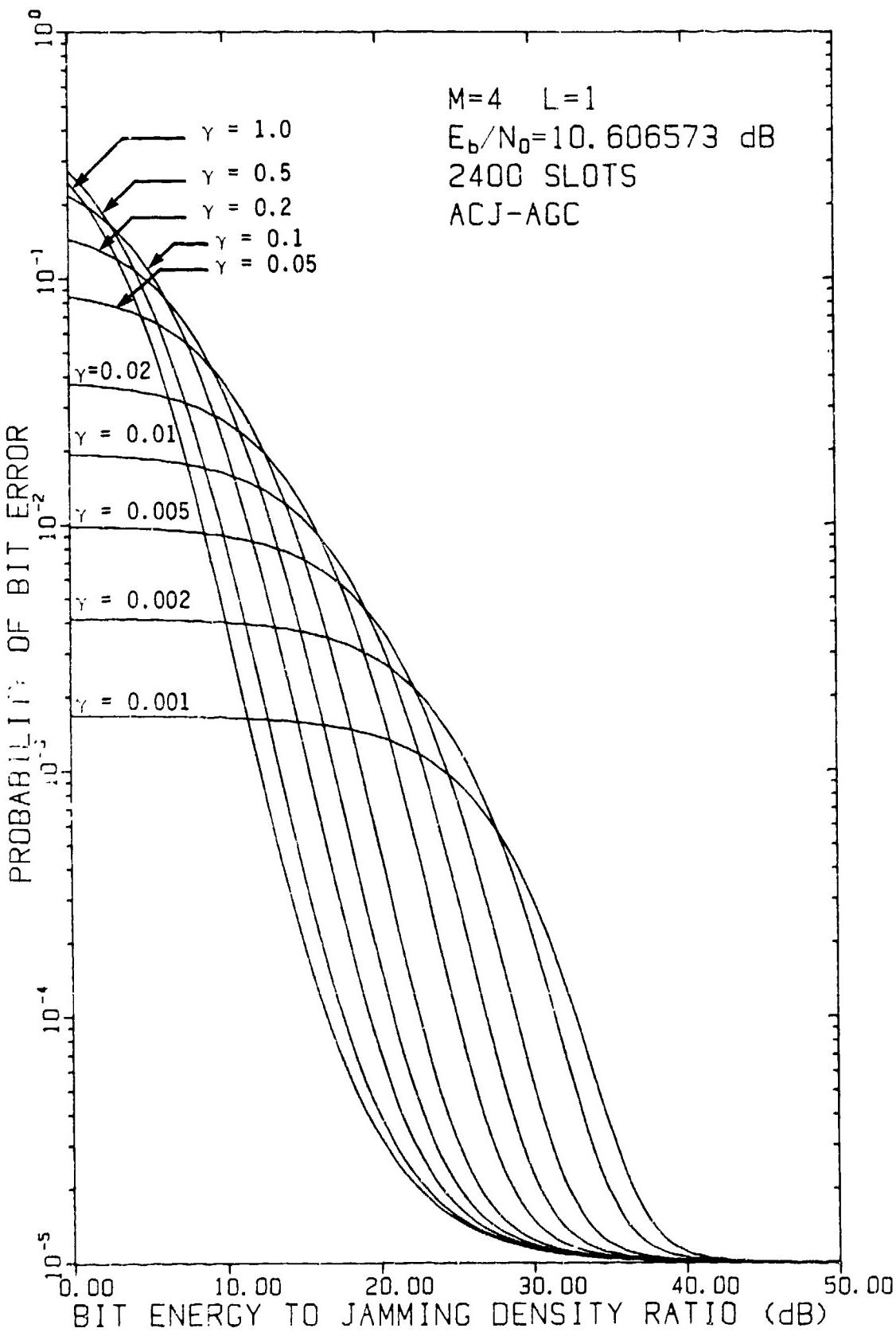


FIGURE 4.4-21 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER WITH $M=4$ AND $L=1$ HOP/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L=1$) WITH JAMMING FRACTION γ AS A PARAMETER

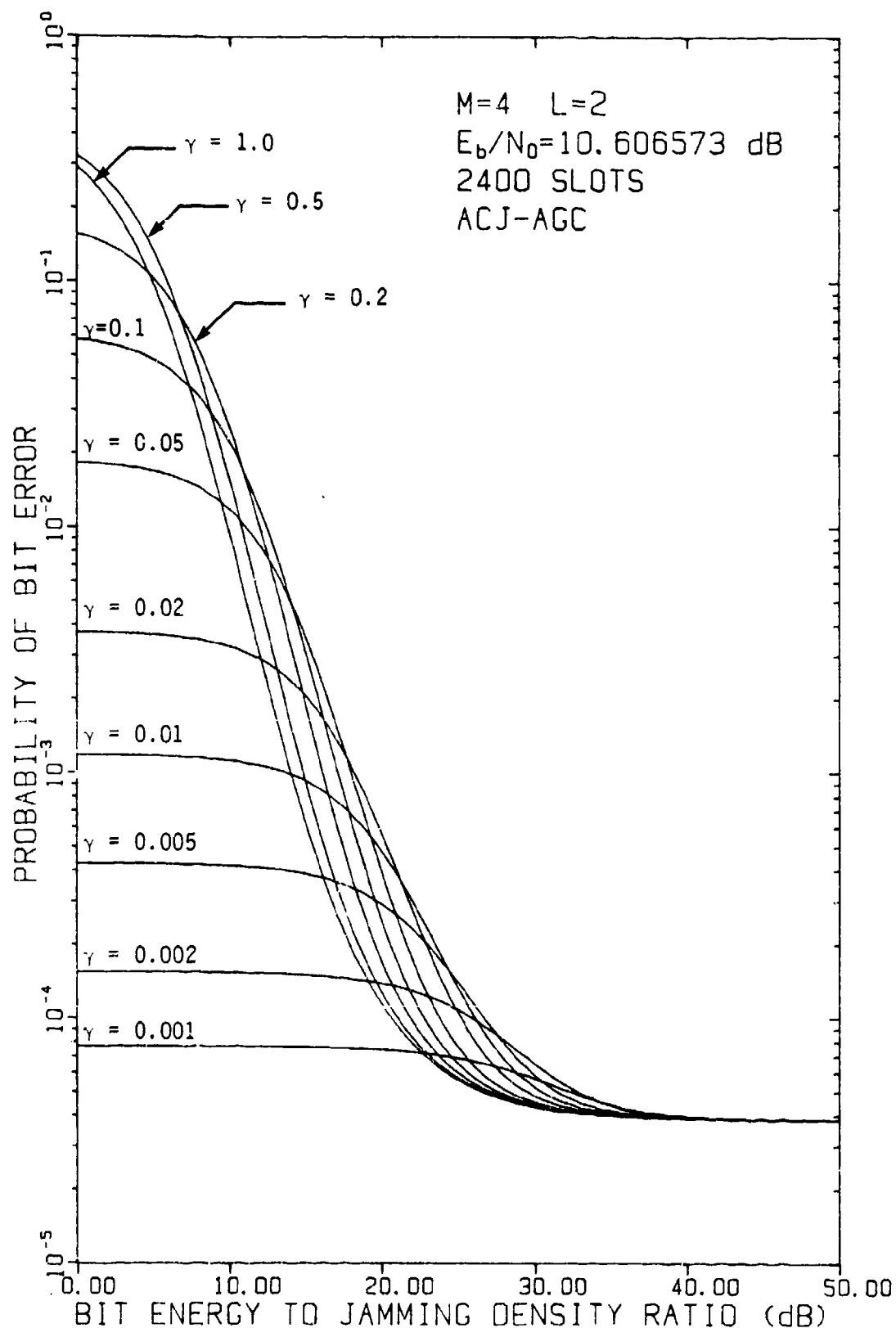


FIGURE 4.4-22 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER WITH $M=4$ AND $L=2$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L=1$) WITH JAMMING FRACTION γ AS A PARAMETER

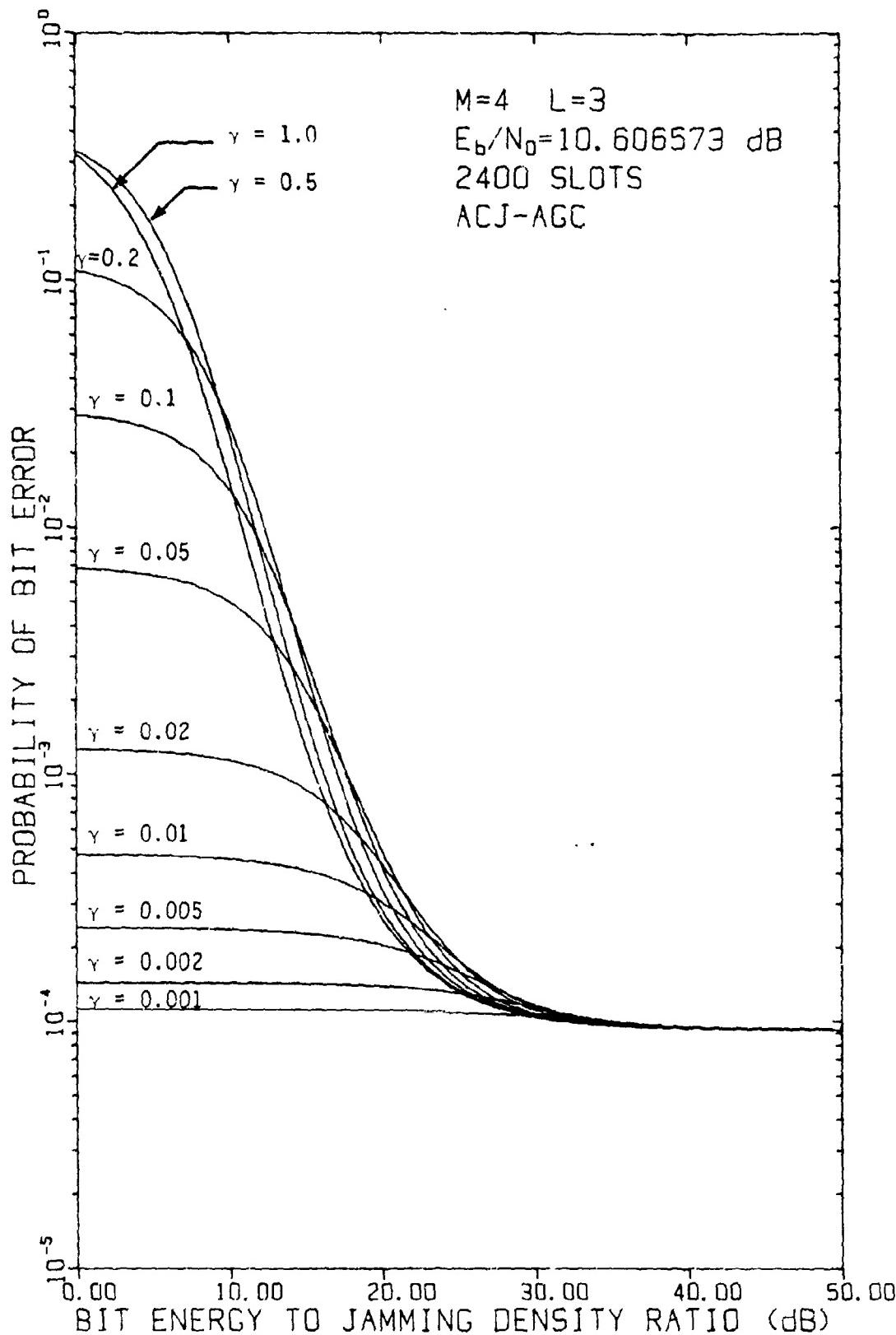


FIGURE 4.4-23 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER WITH $M=4$ AND $L=3$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573$ dB (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L=1$) WITH JAMMING FRACTION γ AS A PARAMETER

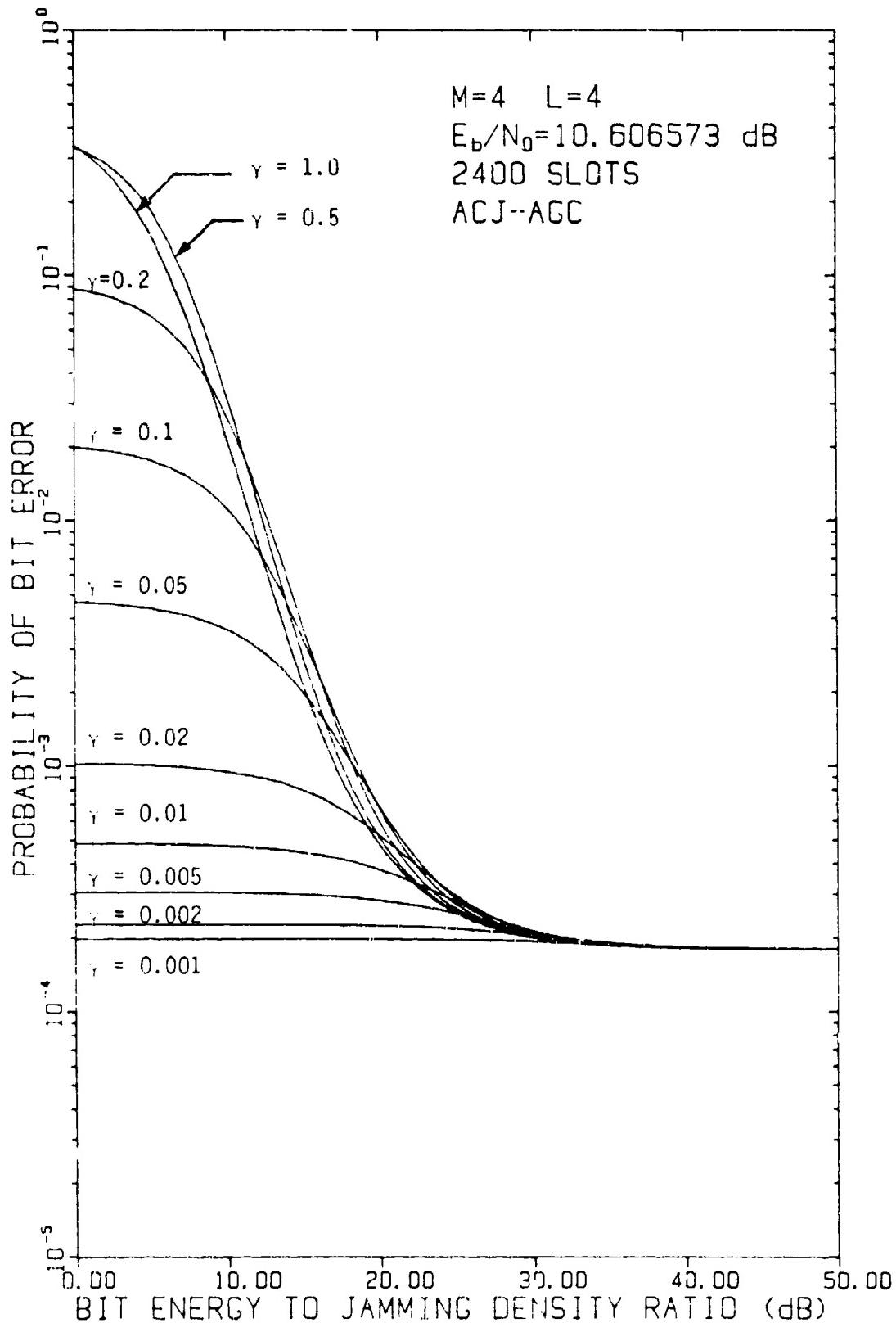


FIGURE 4.4-24 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER WITH $M=4$ AND $L=4$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 10.606573 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L=1$) WITH JAMMING FRACTION γ AS A PARAMETER

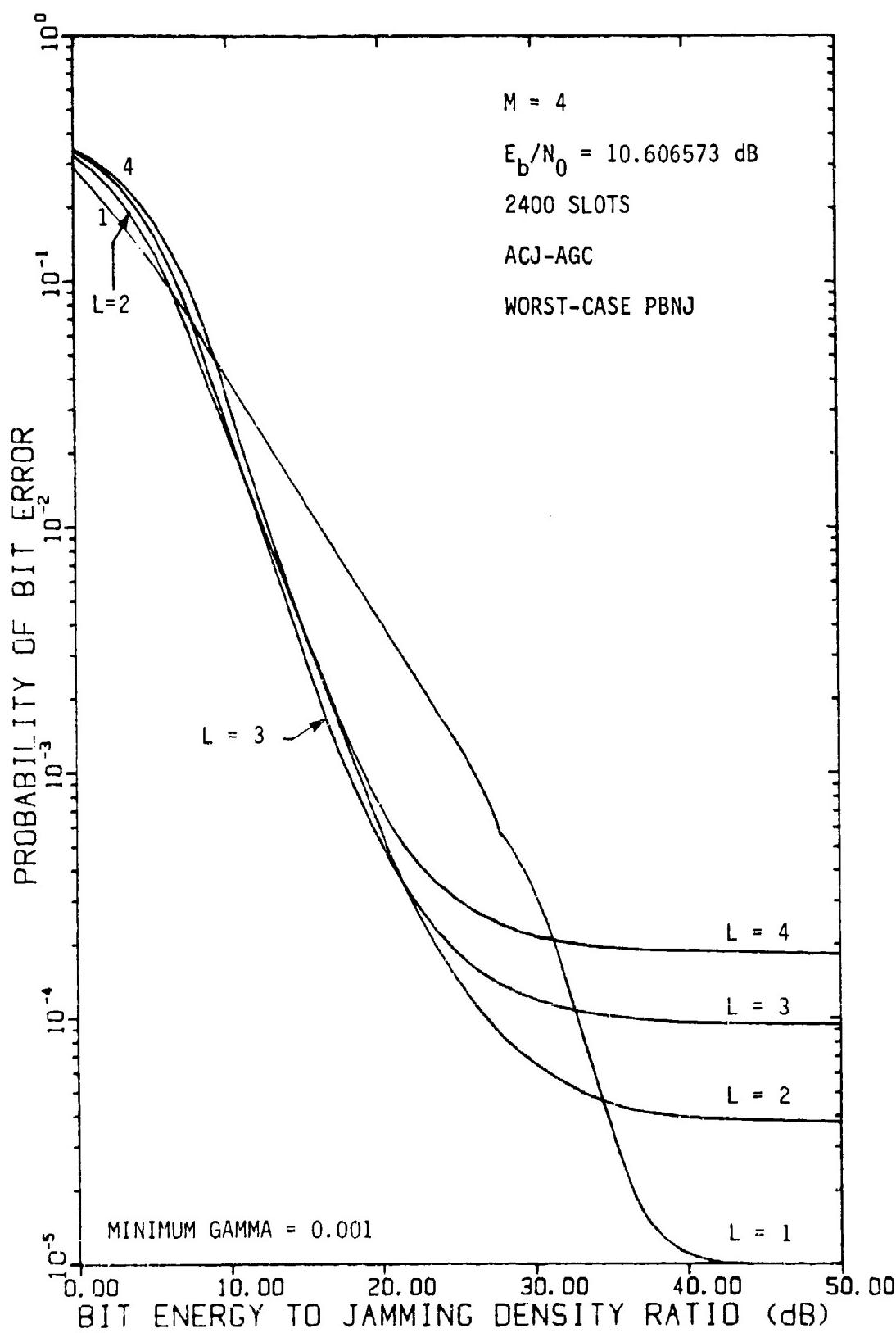


FIGURE 4.4-25 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER AND $M=4$ WITH NUMBER OF HOPS/SYMBOL AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING

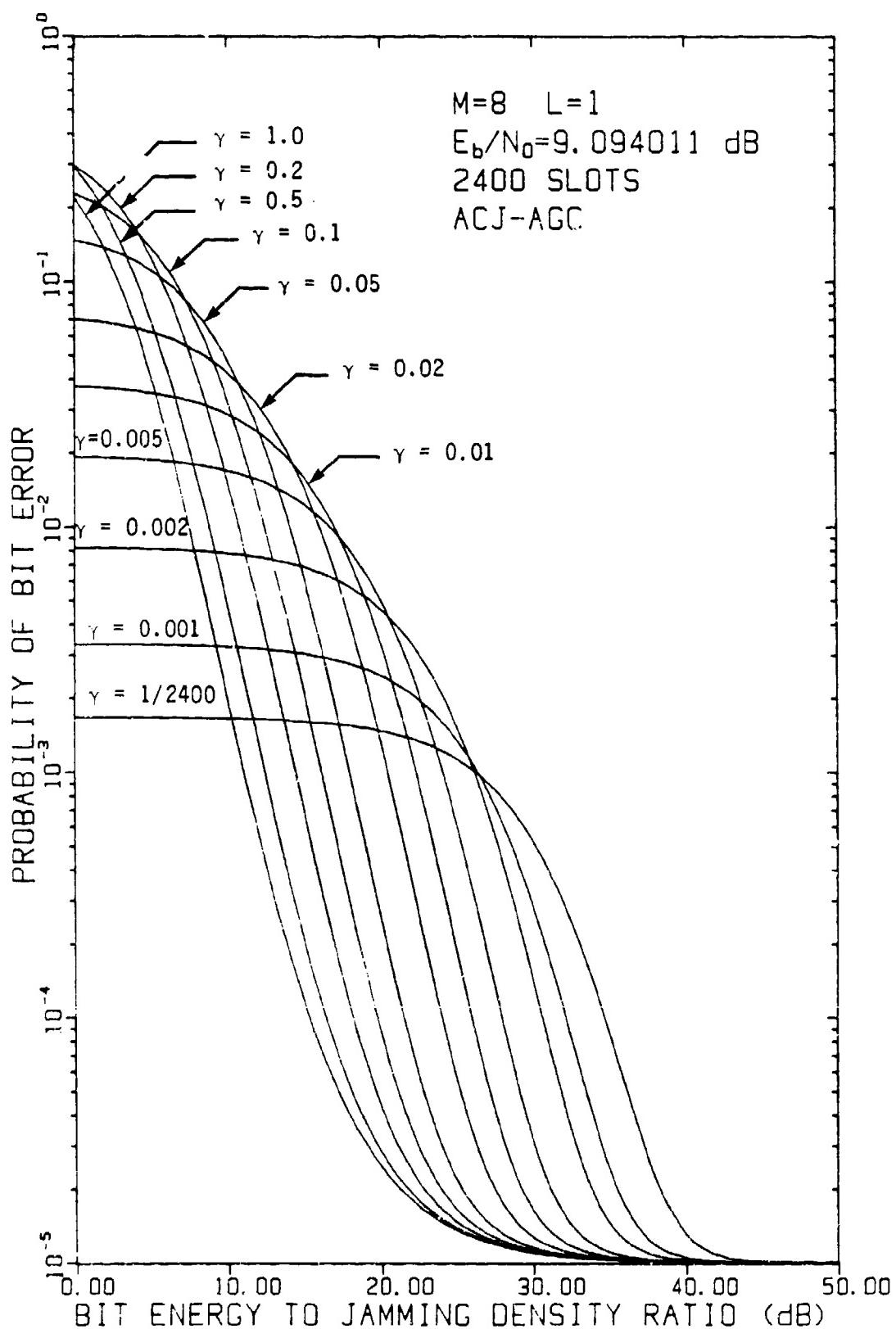


FIGURE 4.4-26 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER WITH $M=8$ AND $L=1$ HOP/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 9.094011 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L=1$) WITH JAMMING FRACTION γ AS A PARAMETER

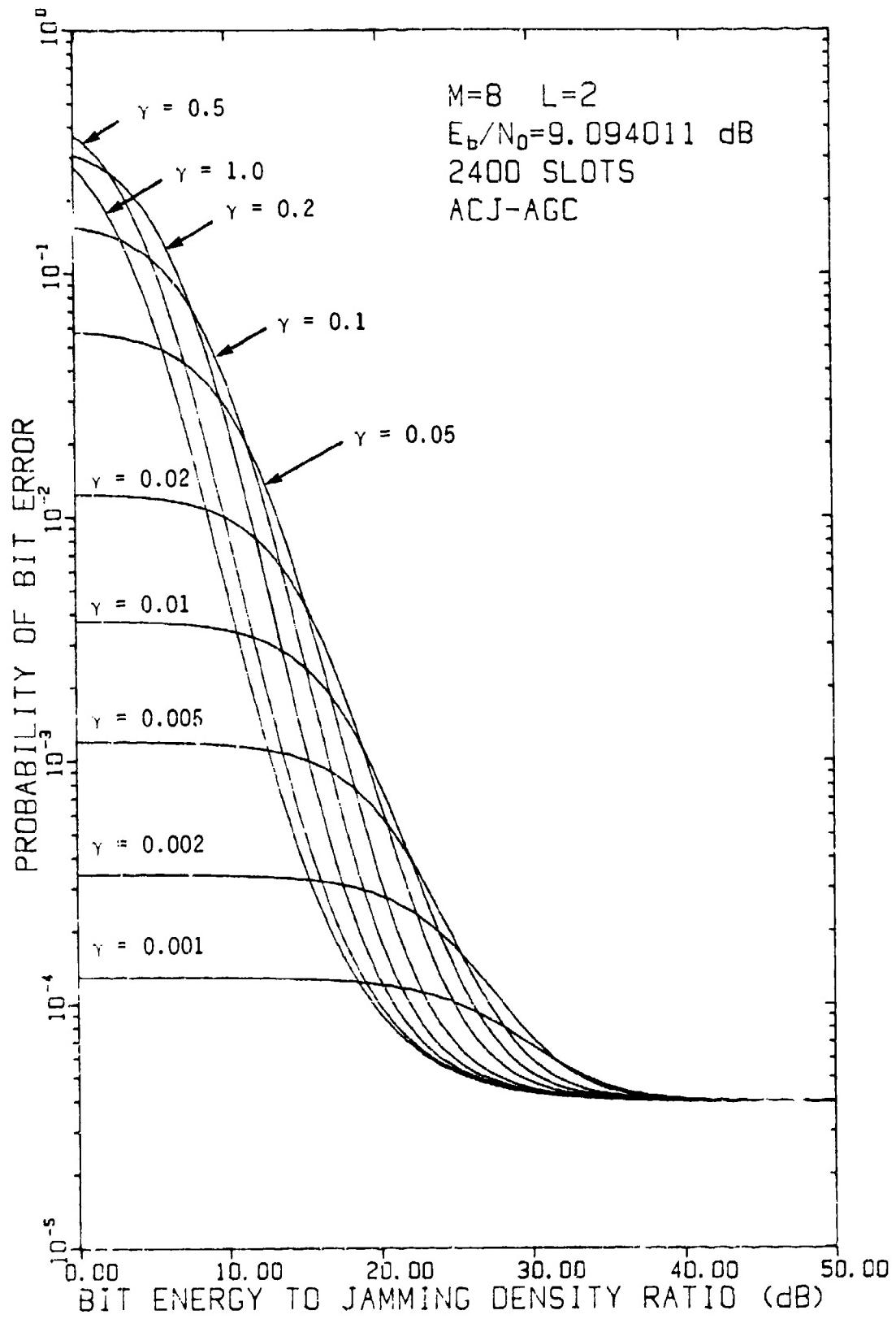


FIGURE 4.4-27 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER WITH $M=8$ AND $L=2$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 9.094011 \text{ dB}$ (FOR 10^{-5} BER WITHOUT JAMMING WHEN $L=1$) WITH JAMMING FRACTION γ AS A PARAMETER

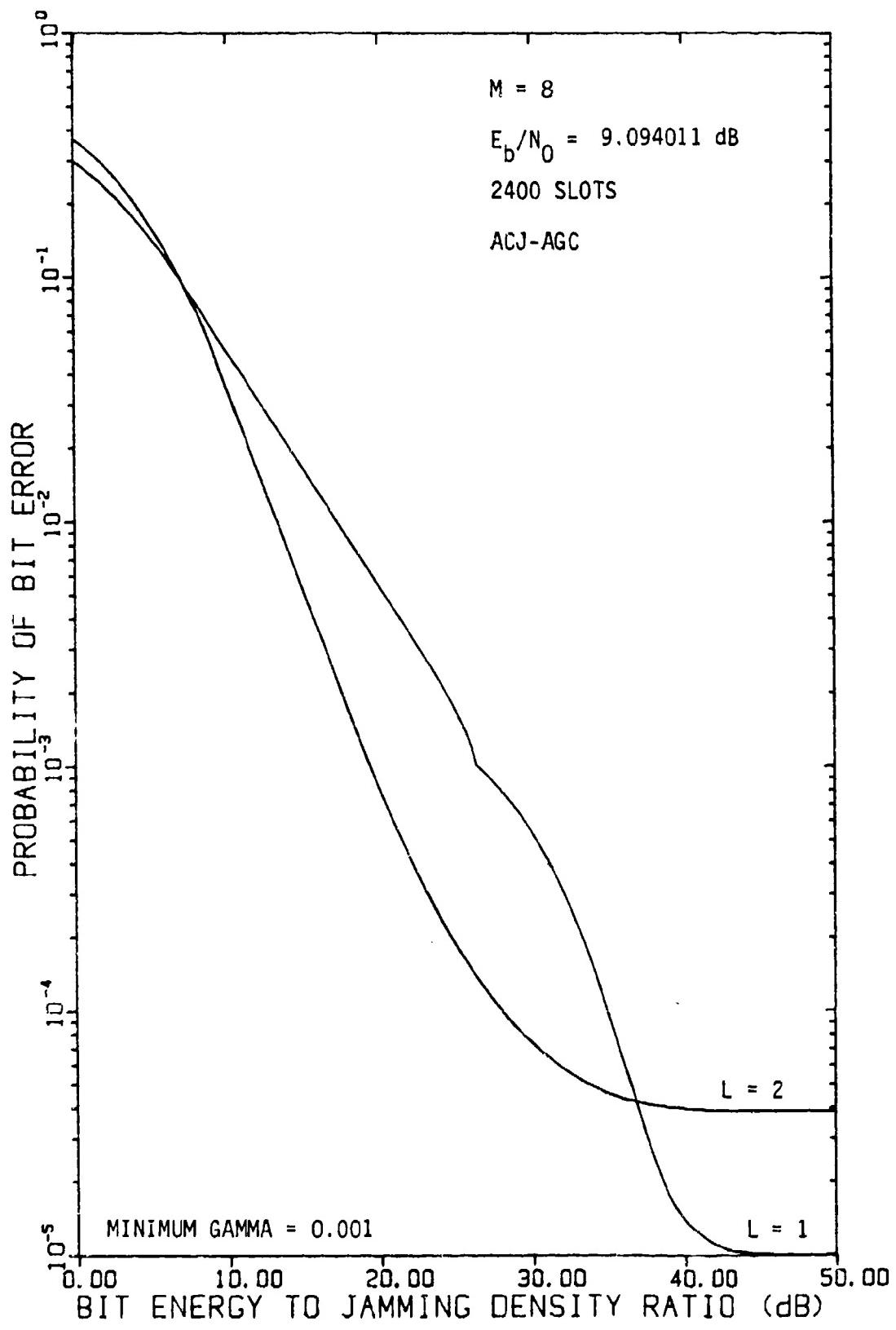


FIGURE 4.4-28 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER AND $M=8$ WITH NUMBER OF HOPS/SYMBOL AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING

J. S. LEE ASSOCIATES, INC.

5.0 FH/RMFSK PERFORMANCE USING CLIPPER RECEIVER

We now undertake analysis of a third type of ECCM receiver for FH/RMFSK, in which the effect of jamming on the symbol decision is reduced by soft-limiting or clipping the per-hop symbol decision variables $\{z_{mk}; m=1, 2, \dots, M; k=1, 2, \dots, L\}$. The receiver structure is diagrammed in Figure 5.0-1. In each of the M dehopped symbol channels, the square-law envelope detector samples are clipped at some level n prior to summing to perform the symbol decision. Because the contribution of a jammed hop to the decision variables is at most n , no matter how strong the jammer noise power, it is expected that an improved performance will result. The clipping threshold n is to be chosen to minimize the error probability when there is no jamming.

In previous analyses of the clipper receiver (for conventional FH/MFSK) we had employed a numerical convolution technique to obtain the distributions of the decision variables. Here we shall obtain the needed probability density functions (pdf's) directly, through analysis.

5.1 DISTRIBUTIONS OF THE DECISION VARIABLES

We first discuss the general form for the pdf of the sum of clipped square-law envelope detector samples, then apply this form to non-signal and signal channels.

5.1.1 General Form of the pdf.

If the input to a clipper with clipping level n has the pdf $f_1(x), x \geq 0$, then the output has the pdf

$$p_1(x) = \begin{cases} f_1(x) + q \delta(x-n) & 0 < x \leq n \\ 0, \text{ otherwise;} & \end{cases} \quad (5.1-1a)$$

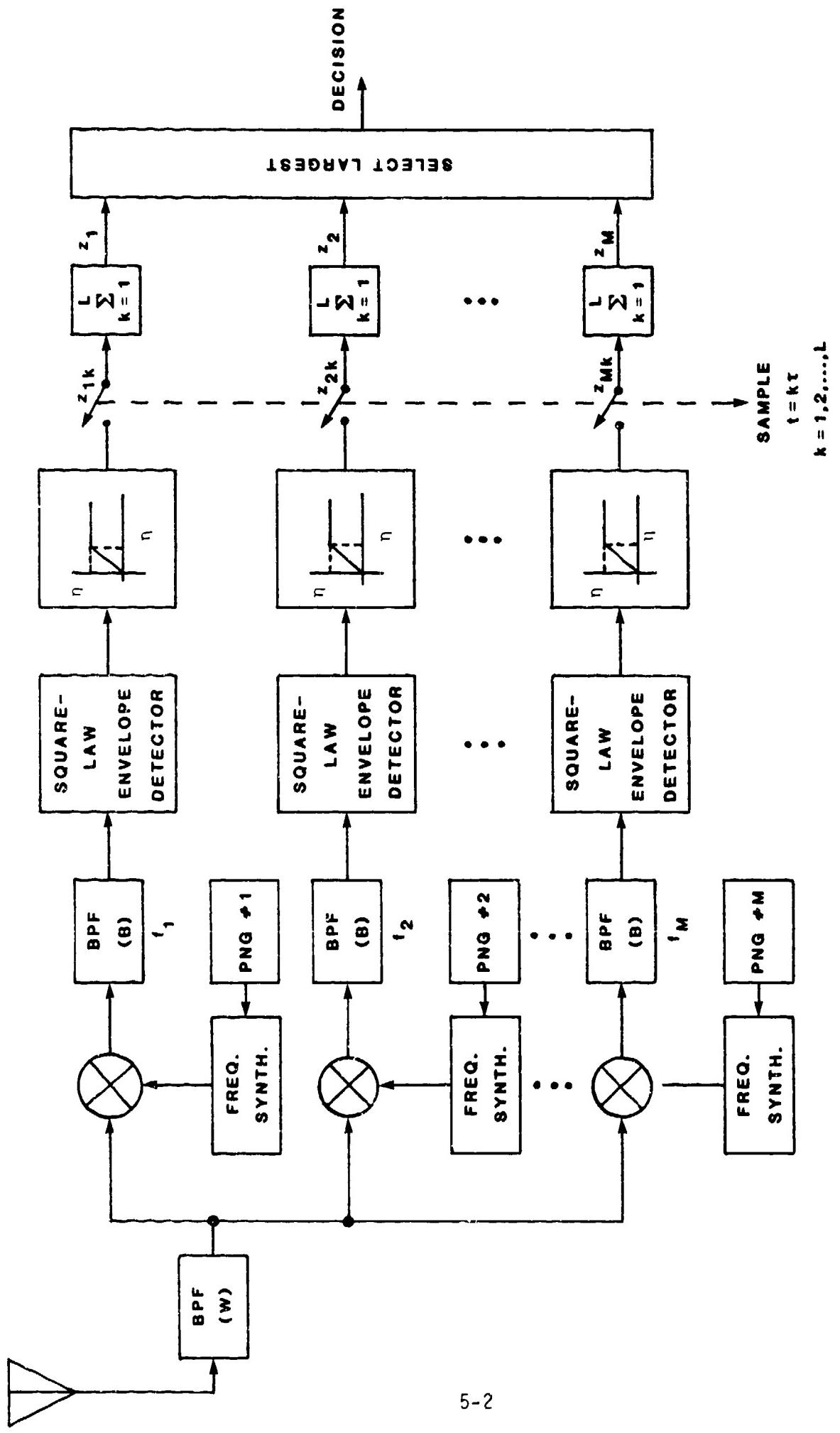


FIGURE 6.0-1 SOFT-DECISION FH/RMFSK RECEIVER WITH CLIPPERS

J. S. LEE ASSOCIATES, INC.

where

$$q = \Pr\{\text{input} > n\} = \int_n^{\infty} d\alpha f_1(\alpha) \quad (5.1-1b)$$

This fact is illustrated by figure 5.1-1.

Now, since individual hops are jammed independently and in any combination, we introduce the notations

$f_L(x;\ell) \equiv$ non-delta function part of the
pdf for the sum of L clipped
samples when ℓ hops in that
channel are jammed. (5.1-2a)

$$q_0 = \Pr\{\text{one sample} > n \mid \text{not jammed}\} \quad (5.1-2b)$$

$$q_1 = \Pr\{\text{one sample} > n \mid \text{jammed}\}. \quad (5.1-2c)$$

Note that it is sufficient to specify only ℓ , the number of hops jammed; the order in which the jamming occurs does not affect the sum. Using this notation, the pdf for a single clipped envelope sample is

$$p_1(x;0) = f_1(x;0) + q_0 \delta(x-n), \text{ hop not jammed}; \quad (5.1-3a)$$

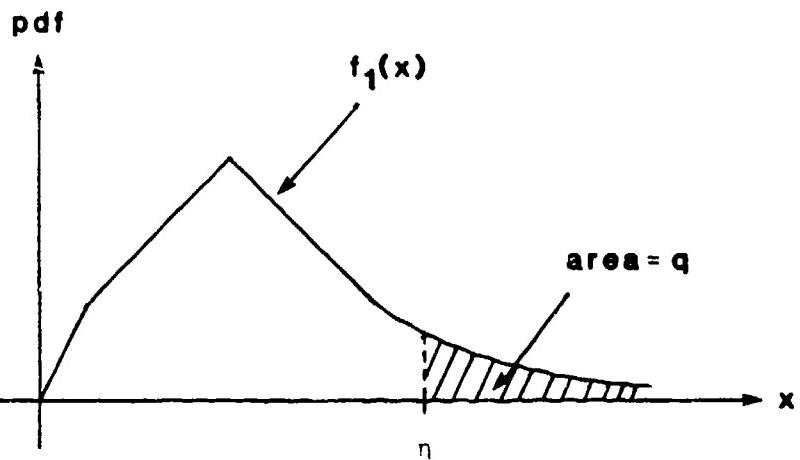
$$p_1(x;1) = f_1(x;1) + q_1 \delta(x-n), \text{ hop jammed}; \quad (5.1-3b)$$

and it is understood that the pdf is zero outside the interval $0 \leq x \leq n$.

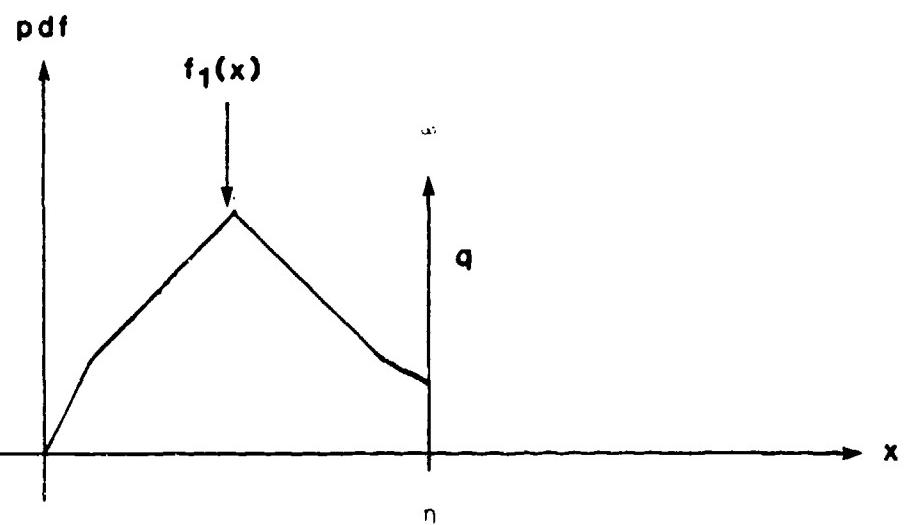
For ℓ hops jammed, the pdf of the sum of L clipped samples can be expressed as the convolution

$$\underbrace{p_1(x;0)*p_1(x;0)*\dots*p_1(x;0)}_{L-\ell \text{ pdf's}} \underbrace{*p_1(x;1)*\dots*p_1(x;1)}_{\ell \text{ pdf's}}. \quad (5.1-4)$$

Thus we have the following general expressions for the sum's pdf for $L=2$ to 4:



(a) pdf before clipping



(b) pdf after clipping

FIGURE 5.1-1 EFFECTS OF CLIPPING ON pdf

J. S. LEE ASSOCIATES, INC.

$$p_1(x;0) = f_2(x;0) + q_0^2 \delta(x-2n), \quad 0 \leq x \leq 2n; \quad (5.1-5a)$$

where

$$f_2(x;0) = \begin{cases} \int_0^x dw f_1(x-w;0) f_1(w;0), & 0 \leq x < n; \\ \int_{x-n}^n dw f_1(x-w;0) f_1(w;0) \\ + 2q_0 f_1(x-n;0), & n \leq x \leq 2n. \end{cases} \quad (5.1-5b)$$

$$p_2(x;1) = f_2(x;1) + q_0 q_1 \delta(x-2n), \quad 0 \leq x \leq 2n; \quad (5.1-6a)$$

where

$$f_2(x;1) = \begin{cases} \int_0^x dw f_1(x-w;1) f_1(w;0), & 0 \leq x < n; \\ \int_{x-n}^n dw f_1(x-w;1) f_1(w;0) \\ + q_0 f_1(x-n;1) + q_1 f_1(x-n;0), & n \leq x \leq 2n. \end{cases} \quad (5.1-6b)$$

$$p_2(x;2) = f_2(x;2) + q_1^2 \delta(x-2n), \quad 0 \leq x \leq 2n; \quad (5.1-7a)$$

where

$$f_2(x;2) = \begin{cases} \int_0^x dw f_1(x-w;1) f_1(w;1), & 0 \leq x < n; \\ \int_{x-n}^n dw f_1(x-w;1) f_1(w;1) \\ + 2q_1 f_1(x-n;1), & n \leq x \leq 2n. \end{cases} \quad (5.1-7b)$$

$$p_3(x;0) = f_3(x;0) + q_0^3 \delta(x-3n), \quad 0 \leq x \leq 3n; \quad (5.1-8a)$$

where

$$f_3(x;0) = \begin{cases} \int_0^x dw f_1(x-w;0) f_2(w;0), & 0 \leq x < n; \\ \int_{x-n}^x dw f_1(x-w;0) f_2(w;0) \\ + q_0 f_2(x-n;0), & n \leq x \leq 2n; \end{cases} \quad (5.1-8b)$$

J. S. LEE ASSOCIATES, INC.

and

$$f_3(x;0) = \begin{cases} \int_{x-\eta}^{2\eta} dw f_1(x-w;0) f_2(w;0) \\ + q_0 f_2(x-\eta;0) \\ + q_0^2 f_1(x-2\eta;0), \quad 2\eta \leq x \leq 3\eta. \end{cases} \quad (5.1-8c)$$

$$p_3(x;1) = f_3(x;1) + q_0^2 q_1 \delta(x-3\eta), \quad 0 \leq x \leq 3\eta; \quad (5.1-9a)$$

where

$$f_3(x;1) = \begin{cases} \int_0^x dw f_1(x-w;1) f_2(w;0), \quad 0 \leq x < \eta \\ \int_{x-\eta}^x dw f_1(x-w;1) f_2(w;0) \\ + q_1 f_2(x-\eta;0), \quad \eta \leq x < 2\eta \\ \int_{x-2\eta}^{2\eta} dw f_1(x-w;1) f_2(w;0) \\ + q_1 f_2(x-\eta;0) \\ + q_0^2 f_1(x-2\eta;1), \quad 2\eta \leq x \leq 3\eta. \end{cases} \quad (5.1-9b)$$

$$p_3(x;2) = f_3(x;2) + q_0 q_1^2 \delta(x-3\eta), \quad 0 \leq x \leq 3\eta; \quad (5.1-10a)$$

J. S. LEE ASSOCIATES, INC.

where

$$f_3(x;2) = \begin{cases} \int_0^x dw f_1(x-w;0)f_2(w;2), & 0 \leq x < \eta; \\ \int_{x-\eta}^x dw f_1(x-w;0)f_2(w;2) \\ + q_0 f_2(x-\eta;2), & \eta \leq x < 2\eta; \\ \int_{x-\eta}^{2\eta} dw f_1(x-w;0)f_2(w;2) \\ + q_0 f_2(x-\eta;2) \\ + q_1^2 f_1(x-2\eta;0), & 2\eta \leq x \leq 3\eta. \end{cases} \quad (5.1-10b)$$

$$p_3(x;3) = f_3(x;3) + q_1^3 \delta(x-3\eta), \quad 0 < x < 3\eta; \quad (5.1-11a)$$

where

$$f_3(x;3) = \begin{cases} \int_0^x dw f_1(x-w;1)f_2(w;2), & 0 \leq x < \eta; \\ \int_{x-\eta}^x dw f_1(x-w;1)f_2(w;2) \\ + q_1 f_2(x-\eta;2), & \eta \leq x < 2\eta; \\ \int_{x-\eta}^{2\eta} dw f_1(x-w;1)f_2(w;2) \\ + q_1 f_2(x-\eta;2) \\ + q_1^2 f_1(x-2\eta;1), & 2\eta \leq x < 3\eta. \end{cases} \quad (5.1-11b)$$

J. S. LEE ASSOCIATES, INC.

$$p_4(x;0) = f_4(x;0) + q_0^4 \delta(x-4\eta), \quad 0 \leq x \leq 4\eta; \quad (5.1-12a)$$

where

$$f_4(x;0) = \begin{cases} \int_0^x dw f_1(x-w;0) f_3(w;0), & 0 \leq x < \eta; \\ \int_{x-\eta}^x dw f_1(x-w;0) f_3(w;0) \\ + q_0 f_3(x-\eta;0), & \eta \leq x < 3\eta \\ \int_{x-\eta}^{3\eta} dw f_1(x-w;0) f_3(w;0) \\ + q_0 f_3(x-\eta;0) \\ + q_0^3 f_1(x-3\eta;0), & 3\eta \leq x \leq 4\eta. \end{cases} \quad (5.1-12b)$$

$$p_4(x;1) = f_4(x;1) + q_0^3 q_1 \delta(x-4\eta), \quad 0 \leq x \leq 4\eta; \quad (5.1-13a)$$

where

$$f_4(x;1) = \begin{cases} \int_0^x dw f_1(x-w;1) f_3(w;0), & 0 \leq x < \eta; \\ \int_{x-\eta}^x dw f_1(x-w;1) f_3(w;0) \\ + q_1 f_3(x-\eta;0), & \eta \leq x < 3\eta; \\ \int_{x-\eta}^{3\eta} dw f_1(x-w;1) f_3(w;0) \\ + q_1 f_3(x-\eta;0) \\ + q_0^3 f_1(x-3\eta;1), & 3\eta < x \leq 4\eta. \end{cases} \quad (5.1-13b)$$

J. S. LEE ASSOCIATES, INC.

$$p_4(x;2) = f_4(x;2) + q_0^2 q_1^2 \delta(x-4n), \quad 0 \leq x \leq 4n; \quad (5.1-14a)$$

where

$$f_4(x;2) = \begin{cases} \int_0^x dw f_2(x-w;2) f_2(w;0), & 0 \leq x < 2n; \\ \int_{x-2n}^{2n} dw f_2(x-w;2) f_2(w;0) \\ + q_0^2 f_2(x-2n;2) \\ + q_1^2 f_2(x-2n;0), & 2n \leq x \leq 4n. \end{cases} \quad (5.1-14b)$$

$$p_4(x;3) = f_4(x;3) + q_0 q_1^3 \delta(x-4n), \quad 0 \leq x \leq 4n; \quad (5.1-15a)$$

where

$$f_4(x;3) = \begin{cases} \int_0^x dw f_1(x-w;0) f_3(w;3), & 0 \leq x < n; \\ \int_{x-n}^x dw f_1(x-w;0) f_3(w;3) \\ + q_0 f_3(x-n;3), & n \leq x < 3n; \\ \int_{x-n}^{3n} dw f_1(x-w;0) f_3(w;3) \\ + q_0 f_3(x-n;3) \\ + q_1^3 f_1(x-3n;0), & 3n \leq x \leq 4n. \end{cases} \quad (5.1-15b)$$

$$p_4(x;4) = f_4(x;4) + q_1^4 \delta(x-4n), \quad 0 \leq x \leq 4n; \quad (5.1-16a)$$

J. S. LEE ASSOCIATES, INC.

$$f_4(x;4) = \begin{cases} \int_0^x dw f_1(x-w;1)f_3(w;3), & 0 \leq x < \eta; \\ \int_{x-\eta}^x dw f_1(x-w;1)f_3(w;3) \\ + q_1 f_3(x-\eta;3), & \eta \leq x < 3\eta; \\ \int_{x-\eta}^{3\eta} dw f_1(x-w;1)f_3(w;3) \\ + q_1 f_3(x-\eta;3) \\ + q_1^3 f_1(x-3\eta;1), & 3\eta \leq x \leq 4\eta. \end{cases} \quad (5.1-16b)$$

5.1.2 Non-Signal Channel pdf.

Assuming without loss of generality that the received signal power S is present in the first ($m=1$) of M dehopped symbol frequency channels, the remaining channels ($m=2,3,\dots,M$) contain only background noise and possibly jamming noise. The samples of the square-law envelope detectors in these channels are independent chi-squared random variables with two degrees of freedom, multiplied by σ_{mk}^2 , where

$$\sigma_{mk}^2 = \begin{cases} \sigma_N^2 = N_0 B, & \text{hop not jammed} \\ \sigma_T^2 = (N_0 + N_J/\gamma)B, & \text{hop jammed.} \end{cases} \quad (5.1-17)$$

J. S. LEE ASSOCIATES, INC.

Consequently, the pdf of the unclipped samples is

$$f_1(x) = \frac{1}{2\sigma_m^2} e^{-x/2\sigma_m^2}, \quad x \geq 0;$$

$$= \begin{cases} a e^{-ax}, & x \geq 0, \text{ hop not jammed;} \\ b e^{-bx}, & x \geq 0, \text{ hop jammed;} \end{cases} \quad (5.1-18a)$$

using

$$a = 1/2\sigma_N^2, \quad b = 1/2\sigma_T^2. \quad (5.1-18b)$$

Also, we have from (5.1-2b and c)

$$q_0 = e^{-an}, \quad q_1 = e^{-bn}. \quad (5.1-19)$$

In order to distinguish the non-signal channel pdf's from that of the signal channel, we adopt the notation

$$g_L(x;\ell) = f_L(x;\ell, S=0) \quad (5.1-20)$$

for the non-delta function part of the pdf of the sum of L clipped samples when ℓ hops in that channel are jammed. Thus we have for channels $\{m:m>2\}$, the sum pdf

$$p_{z_m}(x) = \begin{cases} g_L(x;\ell_m) + (q_0)^{L-\ell_m} (q_1)^{\ell_m} \delta(x-Ln), & 0 \leq x \leq Ln; \\ 0, \text{ otherwise.} \end{cases} \quad (5.1-21)$$

Substituting (5.1-18) in the general convolutional formulas in Section 5.1.1 yields the pdf's listed in Table 5.1-1 for $L=1$ to 3.

TABLE 5.1-1 NON-SIGNAL CHANNEL PROBABILITY DENSITY FUNCTIONS, $L = 1$ TO 3

L	λ	$p_{zm}(x; \lambda)$
1	0	$a e^{-ax} + e^{-an} \delta(x-n), 0 \leq x \leq n; 0, \text{ otherwise.}$
1	1	$b e^{-ax} + e^{-bn} \delta(x-n), 0 \leq x \leq n; 0, \text{ otherwise.}$
2	0	$a^2 x e^{-ax}, 0 \leq x < n; [2a + a^2(2n-x)]e^{-ax} + e^{-2an} \delta(x-2n), n \leq x \leq 2n; 0, \text{ otherwise}$
1	1	$\frac{ab}{a-b} (e^{-bx} - e^{-ax}), 0 < x < n; \frac{a^2}{a-b} e^{-bn-a(x-n)} - \frac{b^2}{a-b} e^{-an-b(x-n)} + e^{-(a+b)n} \delta(x-2n), n \leq x \leq 2n; 0, \text{ otherwise}$
2	2	same as for $\lambda = 0$, but with a replaced by b
3	0	$\frac{1}{2} a^3 x^2 e^{-ax}, 0 \leq x < n; 0, x > 3n, x < 0$ $a^2 [\frac{1}{2} a n^2 + (3+an)(x-n) - a(x-n)^2] e^{-ax}, n \leq x < 2n$ $a[3+3an+\frac{1}{2}a^2n^2 - a(3+an)(x-2n) + \frac{1}{2}a^2(x-2n)^2] e^{-ax} + e^{-3an} \delta(x-3n), 2n \leq x \leq 3n.$
1	1	$\frac{a^2 b}{(a-b)x} \left\{ e^{-bx} - e^{-ax} [1+(a-b)x] \right\}, 0 \leq x < n; 0, x > 3n, x < 0;$ $\frac{a}{a-b} e^{-a(x-n)} \left\{ a \left[\frac{b}{a-b} + a(x-n) \right] e^{-bn} + b \left[\frac{a}{a-b} - (2+an)+a(x-n) \right] e^{-an} \right\} - \frac{2ab^2}{(a-b)x} e^{-an-b(x-n)}, n \leq x < 2n$ $\frac{a^2}{a-b} e^{-(a+b)n-a(x-2n)} \left[\frac{2a-3b}{a-b} + an-a(x-2n) \right] + \frac{b^3}{(a-b)x} e^{-2an-b(x-2n)} + e^{-(2a+b)n} \delta(x-3n), 2n \leq x < 3n.$
2	2	same as for $\lambda = 1$, but with a and b exchanged
3	3	same as for $\lambda = 0$, but with a replaced by b

J. S. LEE ASSOCIATES, INC.

In the conditional probability of error expression,

$$P_s(e|\ell, L) = 1 - E_{z_1} \left\{ \prod_{m=2}^M \Pr\{z_m < z_1 | \ell, L\} \right\}, \quad (5.1-22)$$

the cumulative distribution function for the non-signal channels is needed, written

$$G_L(x; \ell) \triangleq \Pr\{z_m \leq x | \ell, L\}. \quad (5.1-23)$$

This function is given in Table 5.1-2 for $L=1$ to 3.

5.1.3 Signal Channel pdf.

The samples of the square-law envelope detector in the signal channel are independent noncentral chi-squared random variables with two degrees of freedom, multiplied by σ_{1k}^2 , and with noncentrality parameters

$$\lambda_k = 2S/\sigma_{1k}^2 = \begin{cases} 2S/\sigma_N^2 = 2\rho_N, & \text{hop not jammed} \\ 2S/\sigma_T^2 = 2\rho_T, & \text{hop jammed.} \end{cases} \quad (5.1-24)$$

Consequently the pdf of the unclipped samples is

$$\begin{aligned} f_1(x) &= \frac{1}{2\sigma_{1k}^2} e^{-(x+2S)/2\sigma_{1k}^2} I_0(\sqrt{2Sx/\sigma_{1k}^2}) \\ &= \begin{cases} a e^{-a(x+2S)} I_0(2a\sqrt{2Sx}), & x \geq 0, \text{ hop not jammed;} \\ b e^{-b(x+2S)} I_0(2b\sqrt{2Sx}), & x \geq 0, \text{ hop jammed;} \end{cases} \end{aligned} \quad (5.1-25)$$

where a and b are given by (5.1-18b). To distinguish the signal case from the non-signal case, the q_0 and q_1 defined by (5.1-2) will be written in the upper case; the values are

TABLE 5.1-2 NON-SIGNAL CHANNEL CUMULATIVE DISTRIBUTION FUNCTIONS, L=1 TO 3

L	ϱ	$G_L(x; \varrho) = \Pr\{Z_m \leq x \varrho, L\}, m \geq 2$
1	0	$1 - e^{-ax}, 0 \leq x < \eta; 1, x \geq \eta.$ $a = 1/2\sigma_N^2$
1	1	$1 - e^{-bx}, 0 \leq x < \eta; 1, x \geq \eta.$ $b = 1/2\sigma_T^2$
2	0	$1 - e^{-ax}(1+ax), 0 \leq x < \eta; 1 - e^{-ax}[1+a(2\eta-x)], \eta \leq x < 2\eta; 1, x \geq 2\eta.$
1	1	$1 - \frac{1}{a-b}[ae^{-bx} - be^{-ax}], 0 \leq x < \eta; 1 - \frac{1}{a-b}[ae^{-bn-a(x-\eta)} - be^{-an-b(x-\eta)}], \eta \leq x < 2\eta; 1, x \geq 2\eta.$
2	0	$1 - e^{-bx}(1+bx), 0 \leq x < \eta; 1 - e^{-bx}(1+b(2\eta-x)), \eta \leq x < 2\eta; 1, x \geq 2\eta.$
3	0	$1 - e^{-ax}(1+ax+\frac{1}{2}a^2x^2), 0 \leq x < \eta; 1 - e^{-ax}[1+a(2\eta)(x-2\eta)+\frac{1}{2}a^2(x-2\eta)^2], 2\eta \leq x < 3\eta; 1, x \geq 3\eta.$
1	1	$1 - \frac{a^2}{(a-b)}e^{-bx} + \frac{b}{a-b}e^{-ax}\left[\frac{2a-b}{a-b} + ax\right], 0 \leq x < \eta;$ $1 - e^{-a(x-\eta)} \left\{ e^{-bn} \left[\frac{a^2}{(a-b)^2} + \frac{a^2}{a-b}(x-\eta) \right] + e^{-an} \left[\frac{b^2}{(a-b)^2} + \frac{ab}{a-b}(x-2\eta) \right] \right\} + \frac{2ab}{(a-b)^2} e^{-an-b(x-\eta)}, \eta \leq x < 2\eta;$
1	2	$1 - e^{-(a+b)\eta-a(x-2\eta)} \left[\frac{a(1+a\eta)}{a-b} - \frac{ab}{(a-b)^2} - \frac{a^2}{a-b}(x-2\eta) \right] - \frac{b^2}{(a-b)^2} e^{-2an-b(x-2\eta)}, 2\eta \leq x < 3\eta; 1, x \geq 3\eta.$
3	3	Same as for $\varrho = 1$, but with a and b exchanged
		Same as for $\varrho = 0$, but with a replaced by b

$$\begin{aligned} Q_0 &= \Pr\{z_{1k} > n \mid \text{not jammed}\} \\ &= Q(2\sqrt{aS}, \sqrt{2an}) \end{aligned} \quad (5.1-26a)$$

and

$$\begin{aligned} Q_1 &= \Pr\{z_{1k} > n \mid \text{jammed}\} \\ &= Q(2\sqrt{bS}, \sqrt{2bn}), \end{aligned} \quad (5.1-26b)$$

where $Q(x,y)$ is Marcum's Q-function.

For the sum of clipped samples, the pdf is

$$p_{z_1}(x) = \begin{cases} f_L(x; \ell_1) + (Q_0)^{L-\ell_1} (Q_1)^{\ell_1} \delta(x - Ln), & 0 \leq x \leq Ln; \\ 0, & \text{otherwise.} \end{cases} \quad (5.1-27)$$

Substituting (5.1-25) in the general convolutional formulas in Section 5.1.1 yields the pdf's listed in Table 5.1-3 for $L=1, 2$ and Table 5.1-4 for $L=3$.

5.2 ERROR PROBABILITY FORMULATION

Having the pdf's for the FH/RMFSK clip-and-sum decision variables $\{z_m\}$, we can formulate the probability of error.

5.2.1 Conditional Probability of Error.

The probability of symbol error, conditioned on the jamming event $\underline{\ell} = (\ell_1, \ell_2, \dots, \ell_M)$ where ℓ_m is the number of hops jammed (out of L hops) in channel m , can be expressed as parametric in n , the clipping threshold, by

TABLE 5.1-3 SIGNAL CHANNEL PROBABILITY DENSITY FUNCTIONS, $L = 1, 2$

L	λ	$p_{z_1}(x \lambda, L)$
1	0	$a e^{-a(x+2S)} I_0(2a\sqrt{2xS}) + Q_0 \delta(x-\eta), 0 \leq x \leq \eta.$ $a \equiv 1/2\sigma_N^2, Q_0 = Q(2\sqrt{aS}, \sqrt{2a\eta})$
1	1	$b e^{-b(x+2S)} I_0(2b\sqrt{2xS}) + Q_1 \delta(x-\eta), 0 \leq x \leq \eta.$ $b \equiv 1/2\sigma_N^2, Q_1 = Q(2\sqrt{bS}, \sqrt{2b\eta})$
2	0	$\frac{1}{2}a\sqrt{x/S} e^{-a(x+4S)} I_1(4a\sqrt{Sx}), 0 \leq x < \eta; 0, x > 2\eta, x < 0;$ $2aQ_0 e^{-a(x-\eta+2S)} I_0(2a\sqrt{2S(x-\eta)}) + a^2 e^{-a(x+4S)} \int_{x-\eta}^{\eta} dw I_0(2a\sqrt{2Sw}) I_0(2a\sqrt{2S(x-w)})$ $+ Q_0^2 \delta(x-2\eta), \eta \leq x \leq 2\eta.$
1		$a b e^{-bx-2(a+b)S} \int_0^x dw e^{-(a-b)w} I_0(2a\sqrt{2Sw}) I_0(2b\sqrt{2S(x-w)}), 0 \leq x < \eta; 0, x > 2\eta, x < 0;$ $b Q_0 e^{-b(x-\eta+2S)} I_0(2b\sqrt{2S(x-\eta)}) + a Q_1 e^{-a(x-\eta+2S)} I_0(2a\sqrt{2S(x-\eta)})$ $+ a b e^{-bx-2(a+b)S} \int_{x-\eta}^{\eta} dw e^{-(a-b)w} I_0(2a\sqrt{2Sw}) I_0(2b\sqrt{2S(x-w)}) + Q_0 Q_1 \delta(x-2\eta), \eta \leq x \leq 2\eta.$
2		same as for $\lambda = 0$, but with a replaced by b and Q_0 replaced by Q_1 .

TABLE 5.1-4 SIGNAL CHANNEL PROBABILITY DENSITY FUNCTIONS, $L = 3$

L	ϵ	$P_{z1}(x z, L)$
3	0	$(ax/6S) e^{-a(x+6S)} I_2(2a\sqrt{6Sx}), 0 \leq x < n; 0, x > 3n, x < 0;$ $\frac{3}{2} aQ_0 \sqrt{(x-n)/S} e^{-a(x-n+4S)} I_1(4a\sqrt{S(x-n)}) + \frac{3}{2} a^2 e^{-a(x+6S)} \int_{x-n}^n dw \sqrt{w/S} I_1(4a\sqrt{Sw}) I_0(2a\sqrt{2S(x-w)})$ $+ a^3 e^{-a(x+6S)} \int_n^x dw I_0(2a\sqrt{2S(x-w)}) \int_{w-n}^n dz I_0(2a\sqrt{2Sz}) I_0(2a\sqrt{2S(w-z)}), n \leq x < 2n;$ $3aQ_0^2 e^{-a(x-2n+2S)} I_0(2a\sqrt{2S(x-2n)})$ $+ 3a^2 Q_0 e^{-a(x-n+4S)} \int_{x-n}^{2n} dw I_0(2a\sqrt{2S(x-w)}) I_0(2a\sqrt{2S(w-n)}) + Q_0^3 \delta(x-3n)$ $+ a^3 e^{-a(x+6S)} \int_{x-n}^{2n} dw I_0(2a\sqrt{2S(x-w)}) \int_{w-n}^n dz I_0(2a\sqrt{2Sz}) I_0(2a\sqrt{2S(w-z)}), 2n \leq x \leq 3n.$
1		$\frac{1}{2} ab e^{-bx-2(2a+b)S} \int_0^x dw e^{-(a-b)w} \sqrt{w/S} I_0(2b\sqrt{2S(x-w)}) I_1(4a\sqrt{Sw}), 0 \leq x < n; 0, x > 3n, x < 0;$ $\frac{1}{2} aQ_1 \sqrt{(x-n)/S} e^{-a(x-n+4S)} I_1(4a\sqrt{S(x-n)}) + \frac{1}{2} ab e^{-bx-(2a+b)2S} \int_{x-n}^n dw \sqrt{w/S} I_1(2b\sqrt{2S(x-w)}) e^{-(a-b)w}$ $+ 2abQ_0 e^{-b(x-n)-(a+b)2S} \int_0^{x-n} dw I_0(2a\sqrt{2Sw}) I_0(2b\sqrt{2S(x-n-w)}) e^{-(a-b)w}$ $+ a^2 b e^{-bx-(2a+b)2S} \int_n^x dw e^{-(a-b)w} I_0(2b\sqrt{2S(x-w)}) \int_{w-n}^n dz I_0(2a\sqrt{2Sz}) I_0(2a\sqrt{2S(w-z)}), n \leq x \leq 2n;$ $bQ_0^2 e^{-b(x-2n+2S)} I_0(2b\sqrt{2S(x-2n)}) + 2aQ_0 Q_1 e^{-a(x-2n+2S)} I_0(2a\sqrt{2S(x-2n)})$ $+ a^2 Q_1 e^{-a(x-n+4S)} \int_{x-n}^{2n} dw I_0(2a\sqrt{2S(x-w)}) I_0(2a\sqrt{2S(w-n)})$ $+ 2abQ_0 e^{-bx+a(n-(a+b)2S} \int_{x-n}^{2n} dw I_0(2b\sqrt{2S(x-w)}) I_0(2a\sqrt{2S(w-n)}) e^{-(a-b)w}$ $+ a^2 b e^{-bx-(2a+b)2S} \int_{x-n}^{2n} dw e^{-(a-b)w} I_0(2b\sqrt{2S(x-w)}) \int_{w-n}^n dz I_0(2a\sqrt{2Sz}) I_0(2a\sqrt{2S(w-z)}),$ $2n \leq x \leq 3n.$
2		same as for $z = 1$, but with a and b exchanged and Q_0 and Q_1 exchanged
3		same as for $z = 0$, but with a replaced by b and Q_0 replaced by Q_1

J. S. LEE ASSOCIATES, INC.

$$P_s(e; n | L, \underline{\ell}) = 1 - \Pr\{C \equiv \text{correct decision}; n | L, \underline{\ell}\}. \quad (5.2-1)$$

Since there is clipping, there is a finite probability that one or more of the $\{z_m\}$ are equal to L . Thus an appropriate formulation, assuming a randomized decision rule, is

$$\begin{aligned} \Pr\{C; n | L, \underline{\ell}\} &= \sum_{p=0}^M \Pr\{C \text{ and } (p \text{ channels} = L_n); n | L, \underline{\ell}\} \\ &= \sum_{p=0}^M \Pr\{C \text{ and } (p \text{ channels} = L_n, \text{ including signal channel}); n | L, \underline{\ell}\} \\ &= \Pr\{C \text{ and no channels} = L_n; n | L, \underline{\ell}\} \\ &\quad + \sum_{p=0}^{M-1} \Pr\{C \text{ and } (\text{signal channel} = L_n) \text{ and } (p \text{ non-signal channels} = L_n); \\ &\quad \quad \quad n | L, \underline{\ell}\}. \end{aligned} \quad (5.2-2)$$

The first term in (5.2-2) is

$$\int_0^{L_n} dx f_L(x; \ell_1) \prod_{m=2}^M G_L(x; \ell_m), \quad (5.2-3)$$

where $f_L(x; \ell_1)$ is the non-delta function part of the signal channel's pdf, and $G_L(x; \ell_m)$, $m \geq 2$, is the cumulative distribution function for the non-signal channels. (We assume without loss of generality that the signal channel is the first one, i.e., $m=1$.) The sum in (5.2-2) can be expanded as

$$\begin{aligned} &\sum_{p=0}^{M-1} \Pr\{C; n | (z_1 = L_n) \text{ and } (p \text{ non-signal channels} = L_n); L, \underline{\ell}\} \\ &\quad \cdot \Pr\{p \text{ non-signal channels} = L_n | L, \ell_2, \dots, \ell_M\} \\ &\quad \cdot \Pr\{z_1 = L_n | L, \ell_1\} \\ &= \sum_{p=0}^{M-1} \frac{1}{p+1} \Pr\{p \text{ non-signal channels} = L_n | L, \ell_2, \dots, \ell_M\} \cdot P_{1L}(\ell_1), \end{aligned} \quad (5.2.4a)$$

J. S. LEE ASSOCIATES, INC.

and we use $P_{1L}(\ell_1) \triangleq Q_0^{L-\ell_1} Q_1^{\ell_1}$. (5.2-4b)

Using a similar notation, the probability of a non-signal channel's being equal to L_n , i.e., $z_m = L_n$ for $m \geq 2$, is

$$P_{2L}(\ell_m) \triangleq q_0^{L-\ell_m} q_1^{\ell_m} = e^{-(L-\ell_m)a_n - \ell_m b_n}. \quad (5.2-5)$$

Now, there are $\binom{M-1}{p}$ ways for p of the $M-1$ non-signal channels to be selected as either $z_m = L_n$ or $z_m < L_n$. However, it is necessary to account for the fact that these channels may have different numbers of hops jammed, ℓ_m . Let

$$v_m = \begin{cases} 1 & \text{if } z_m = L_n \\ 0 & \text{if } z_m < L_n; \end{cases} \quad (5.2-6)$$

using this indicator variable, and the vector

$$\underline{v} = (v_2, v_3, \dots, v_M), \quad (5.2-7)$$

we can write

$$\begin{aligned} & \Pr\{p \text{ non-signal channels} = L_n | L, \ell_2, \dots, \ell_M\} \\ &= \sum_{\underline{v}} \prod_{m=2}^M \left\{ v_m P_{2L}(\ell_m) + (1-v_m) [1 - P_{2L}(\ell_m)] \right\} \cdot \delta\left(\sum_m v_m, p\right) \end{aligned} \quad (5.2-8a)$$

where

$$\delta(n, p) \triangleq \begin{cases} 1, & p = n \\ 0, & p \neq n. \end{cases} \quad (5.2-8b)$$

J. S. LEE ASSOCIATES, INC.

For example, if all the non-signal channels have the same number of jammed hops, $\ell_m = \ell$, then (5.2-8a) is evaluated as

$$\begin{aligned} & [P_{2L}(\ell)]^p [1 - P_{2L}(\ell)]^{M-1-p} \sum_v \delta(\sum_m v_m, p) \\ &= \binom{M-1}{p} [P_{2L}(\ell)]^p [1 - P_{2L}(\ell)]^{-1-p}. \end{aligned} \quad (5.2-9)$$

Substituting (5.2-8) and (5.2-3) into the error expression results in

$$\begin{aligned} P_s(e; n | L, \underline{\ell}) &= 1 - \int_0^{L_n} dx f_L(x; \ell_1) \prod_{m=2}^M G_L(x; \ell_m) \\ &- P_{1L}(\ell_1) \sum_{p=0}^{M-1} \frac{1}{1+p} \sum_v \prod_{m=2}^M \left\{ v_m P_{2L}(\ell_m) + (1-v_m) [1 - P_{2L}(\ell_m)] \right\} \delta(\sum_m v_m, p). \end{aligned} \quad (5.2-10)$$

5.2.1.1 Special case: $L=1$ (one hop/symbol).

For $L=1$, we have

$$f_1(x; \ell_1) = c_1 e^{-c_1(x+2s)} I_0(2c_1 \sqrt{2sx}); \quad (5.2-11)$$

using

$$c_m = \begin{cases} a & \text{if } \ell_m = 0 \\ b & \text{if } \ell_m = 1; \end{cases} \quad (5.2-12)$$

$$P_{21}(\ell_m) = e^{-c_m n}; \quad (5.2-13)$$

and

J. S. LEE ASSOCIATES, INC.

$$\begin{aligned}
 \prod_{m=2}^M G_L(x; \ell_m) &= \prod_{m=2}^M (1 - e^{-c_m x}) \\
 &= (1 - e^{-ax})^{n_0} (1 - e^{-bx})^{M-1-n_0} \\
 &= \sum_{k=0}^{n_0} \sum_{r=0}^{M-1-n_0} \binom{n_0}{k} \binom{M-1-n_0}{r} (-1)^{k+r} e^{-(ka+rb)x}
 \end{aligned} \tag{5.2-14a}$$

where

$$n_0 \triangleq \#\{\ell_m = 0, m \geq 2\}, \tag{5.2-14b}$$

that is, n_0 is the number of unjammed, non-signal channels. Substituting in the error expression (5.2-10) results in

$$\begin{aligned}
 P_s(e; n | 1, \underline{\ell}) &= \sum_{k=0}^{n_0} \sum_{\substack{r=0 \\ k+r>0}}^{M-1-n_0} \binom{n_0}{k} \binom{M-1-n_0}{r} (-1)^{k+r+1} \\
 &\quad \cdot \int_0^n dx c_1 e^{-2c_1 s - (c_1 + ka + rb)x} I_0(2c_1 \sqrt{sx}) + Q_{\ell_1} \\
 &\quad - Q_{\ell_1} \sum_{p=0}^{M-1} \frac{1}{1+p} \sum_v \prod_{m=2}^M \left\{ (2v_m - 1) e^{-c_m v_m} + 1 - v_m \right\} \delta(\sum_m v_m, p).
 \end{aligned} \tag{5.2-15}$$

The integral equals

$$\frac{c_1}{c_1 + ka + rb} \exp \left\{ \frac{-(ka+rb)2c_1 s}{c_1 + ka + rb} \right\} \left[1 - Q \left(\sqrt{\frac{4c_1^2 s}{c_1 + ka + rb}}, \sqrt{2(c_1 + ka + rb)s} \right) \right]. \tag{5.2-16}$$

Also, since $\ell_m = 0$ or 1 when $L = 1$, the last term can be written

J. S. LEE ASSOCIATES, INC.

$$- Q(2\sqrt{c_1 s}, \sqrt{2c_1 n}) \sum_{p=0}^{M-1} \frac{1}{1+p} \sum_{p_0=p_{\min}}^{p_{\max}} \binom{n_0}{p_0} \binom{M-1-n_0}{p-p_0} e^{-p_0(a-b)n-pbn} \\ \times (1-e^{-an})^{n_0-p_0} (1-e^{-bn})^{M-1-n_0-(p-p_0)}, \quad (5.2-17)$$

where

$$p_{\min} = \max[0, p - (M-1-n_0)], \quad p_{\max} = \min(p, n_0). \quad (5.2-17)$$

For example, if $M=2$, (5.2-15) becomes, since $n_0 = 0$ or 1
 $(n_0 \equiv 1-\ell_2)$,

$$p_b(e; n | 1, \underline{\ell}) = \ell_2 \cdot \frac{c_1}{c_1+b} \exp\left\{-\frac{2bc_1s}{c_1+b}\right\} \left[1 - Q\left(\sqrt{\frac{4c_1^2 s}{c_1+b}}, \sqrt{2(c_1+b)n}\right) \right] \\ + (1-\ell_2) \cdot \frac{c_1}{c_1+a} \exp\left\{-\frac{2ac_1s}{c_1+a}\right\} \left[1 - Q\left(\sqrt{\frac{4c_1^2 s}{c_1+a}}, \sqrt{2(c_1+a)n}\right) \right] \\ - Q(2\sqrt{c_1 s}, \sqrt{2c_1 n}) \left\{ \ell_2 \cdot \left[1 - e^{-bn} + \frac{1}{2} e^{-bn} \right] \right. \\ \left. + (1-\ell_2) \cdot \left[1 - e^{-an} + \frac{1}{2} e^{-an} \right] \right\} \\ + Q(2\sqrt{c_1 s}, \sqrt{2c_1 n}) \quad (5.2-18)$$

5.2.1.2 Special case: $\underline{\ell} = 0$ (no jamming).

For this case, we substitute $\underline{\ell} = 0$ in (5.2-10) to get

J. S. LEE ASSOCIATES, INC.

$$P_s(e; n | L, \underline{C}) = 1 - \int_0^{L_n} dx f_L(x; 0) [G_L(x; 0)]^{M-1} \\ - Q_0^L \sum_{p=0}^{M-1} \frac{1}{1+p} \binom{M-1}{p} e^{-L_p a n} (1-e^{-L_p a n})^{M-1-p} \quad (5.2-19a)$$

$$= \sum_{k=1}^{M-1} \binom{M-1}{k} (-1)^{k+1} \int_0^{L_n} dx f_L(x; 0) [1-G_L(x; 0)]^k \\ + Q_0^L \cdot \frac{e^{L_n a n}}{M} \left[1 - (1-e^{-L_n a n})^M \right]. \quad (5.2-19b)$$

Now, $1-G_L(x; 0)$ in the integral has the form (see Tables 5.1-3 and 5.1-4)

$$1-G_L(x; 0) = \begin{cases} 1, & x < 0; \\ e^{-ax} h_r(x-rn), & rn \leq x < (r+1)n, r=0, 1, 2, \dots, L-1; \\ 0, & x > L_n; \end{cases} \quad (5.2-20)$$

where $h_r(x)$ is an $(L-1)$ degree polynomial. Using this form, the integral in (5.2-19) becomes

$$\sum_{r=0}^{L-1} \int_{rn}^{(r+1)n} dx f_L(x; 0) [h_r(x-rn)]^k e^{-kax} \\ = \sum_{r=0}^{L-1} \int_0^n dx f_L(x+rn; 0) [h_r(x)]^k e^{-ka(x+rn)}. \quad (5.2-21)$$

J. S. LEE ASSOCIATES, INC.

Noting also from Tables 5.1-5 and 5.1-6 that the signal channel pdf can be written

$$f_L(x;0) = e^{-ax} v_r(x-m), \quad r=0,1,\dots,L-1, \quad (5.2-2)$$

we further manipulate (5.2-21) to obtain

$$\sum_{r=0}^{L-1} e^{-(k+1)ra} \int_0^n dx e^{-(k+1)ax} v_r(x) [h_r(x)]^k. \quad (5.2-23)$$

For example, if $L=1$, then $h_0(x) \equiv 1$ and
 $v_0(x) = a e^{-2ax} I_0(2a\sqrt{2Sx})$, giving for (5.2-23) the value

$$\frac{1}{1+k} \exp \left\{ -\frac{2kaS}{k+1} \right\} \left[1-Q \left(\sqrt{\frac{4aS}{1+k}}, \sqrt{2a(k+1)n} \right) \right] \quad (5.2-24)$$

and

$$P_s(e;n|1,0) = \sum_{k=1}^{M-1} \binom{M-1}{k} \frac{(-1)^{k+1}}{k+1} \exp \left\{ \frac{-2kaS}{1+k} \right\} \left[1-Q \left(\sqrt{\frac{4aS}{1+k}}, \sqrt{2a(k+1)n} \right) \right].$$

$$+ Q_0 - Q_0 \frac{e^{an}}{M} \left[1 - (1-e^{-an})^M \right]. \quad (5.2-25)$$

5.2.2 Total Probability of Error.

For a given number of hops/symbol, L , the total symbol probability of error is

$$P_s(e;n,L) = \sum_{\underline{\ell}} \Pr\{\underline{\ell}\} P_s(e;n|L,\underline{\ell}); \quad (5.2-26)$$

J. S. LEE ASSOCIATES, INC.

the bit error probability is

$$P_b(e; n, L) = \frac{M/2}{M-1} P_s(e; n, L). \quad (5.2-27)$$

5.2.2.1 Choice of clipping threshold.

The procedure we have adopted for choosing n , the clipping threshold, is the following: choose n to minimize the error probability when there is no jamming. That is,

$$n^* : \min_n P_s(e; n | L, 0). \quad (5.2-28)$$

Differentiation of the error expression (5.2-19a) gives an equation for the optimum n thus defined. This equation may be written

$$\begin{aligned} & - f_L(Ln; 0) (1 - e^{-Lan})^{M-1} \\ & + \sum_{r=1}^{L-1} \int_0^n dx \frac{\partial}{\partial n} \left\{ f_L(x+rn; 0) [G_L(x+rn; 0)]^{M-1} \right\} \\ & + L Q_0 f_1(n; 0) \frac{e^{Lan}}{M} \left[1 - (1 - e^{-Lan})^M \right] \\ & - Q_0^L \frac{La e^{Lan}}{M} \left\{ 1 - (1 - e^{-Lan})^M - M e^{-Lan} (1 - e^{-Lan})^{M-1} \right\} \end{aligned} \quad (5.2-29)$$

For $L=1$, the second term is zero and the equation can be put in the form

$$\frac{e^{an}}{M} \left[Q_0 a - f_1(n; 0) \right] \left\{ [1 + (M-1)e^{-an}] (1 - e^{-an})^{M-1} - 1 \right\}; \quad (5.2-30)$$

J. S. LEE ASSOCIATES, INC.

this partial derivative with respect to n is negative, indicating the error decreases as n increases, indefinitely*. Thus for $L=1$ the optimum threshold is infinite (no clipping):

$$n^*(L=1) \rightarrow \infty . \quad (5.2-31)$$

For $L>1$, it is not feasible to find the optimum threshold by differentiation; it must be done numerically.

5.2.2.2 Total error for $L=1$.

Since the optimum threshold for $L=1$ is $n^* \rightarrow \infty$, we may express the total error probability by using (5.2-15) to obtain

$$P_b(e; L=1) = \frac{M/2}{M-1} \sum_{\underline{\ell}} \Pr\{\underline{\ell}\} \sum_{k=0}^{n_0} \sum_{r=0}^{M-1-n_0} \binom{n_0}{k} \binom{M-1-n_0}{r} (-1)^{k+r+1} \\ \cdot \frac{c_1}{c_1 + ka + rb} \exp\left\{-\frac{(ka+rb)2c_1S}{c_1 + ka + rb}\right\}, \quad (5.2-32a)$$

$$\text{where } c_1 = (1-\ell_1)a + \ell_1 b \quad (5.2-32b)$$

$$\text{and } n_0 = M-1 - \sum_{m=2}^M \ell_m. \quad (5.2-32c)$$

This, of course, gives exactly the same performance as the other receiver processing schemes for $L=1$.

* The last factor in (5.2-30) can be recognized as the quantity

$-\sum_{m=2}^M \binom{M}{m} (1-e^{-an})^{n-m} e^{-man} < 0$. The second factor is always positive since $aQ_0 = f_1(n; 0) + \exp\{-a(2S+n)\} \sum_{k>1} a(2S/n)^k I_k(2a\sqrt{Sn})$.

J. S. LEE ASSOCIATES, INC.

5.3 NUMERICAL RESULTS

In this section we present some numerical results for the clipper receiver's performance. These results are less voluminous than those obtained for other receivers because of the extremely long computer run times for the clipper equations.*

Two stages of computation are required for the clipper receiver. First, the optimum clipping level in the absence of jamming, η_0/σ_N^2 , must be found by a numerical search. Then this value must be used in computing the jammed performance. Whenever L , M , or E_b/N_0 changes the optimum clipping level must be recomputed.

The many numerical integrations required to evaluate (5.2-10) using the forms for $f_L(x;\ell_1)$ from Tables 5.1-3 and 5.1-4 and for $G_L(x;\ell)$ from Table 5.1-2 result in very lengthy computations. Consider, for example, the case of $M=4$, $L=2$, in which the numerical integrations which are required have the structure

$$1 - \left[\int f_1 g + \int (f_2 + \int f_3) g \right], \quad \ell_1 = 0 \text{ or } \ell_1 = 2 \quad (5.3-1a)$$

$$1 - \left[\int f_1 \int f_2 g + \int (f_3 + \int f_4) g \right], \quad \ell_1 = 1 \quad (5.3-1b)$$

where g is a function of ℓ_2 . Each conditional error probability involves one or two double integrations which must be evaluated numerically to sufficient accuracy as to leave several significant digits after subtracting from 1. This subtractive cancellation problem is especially severe for high E_b/N_j when $P(e)$ is small.

For $L=3$ the situation is even worse, for the numerical integrations take the forms

*It has been noted that for $L=1$ and any M value, the clipper receiver with optimum threshold is merely a conventional receiver, since that threshold is infinite for $L>1$. Thus the results computed previously for $L=1$ apply to this Section as well.

$$1 - \left\{ \int f_1 g + \int [f_1 + \int f_2 + \int (f_3 \int f_4)] g + \int [f_5 + \int f_6 + \int (f_7 \int f_8)] g \right\},$$

$$\ell_1 = 0 \text{ or } \ell_1 = 3 \quad (5.3-2a)$$

$$1 - \left\{ \int (\int f_1) g + \int [f_2 + \int f_3 + \int f_4 + \int f_5 (\int f_6)] g \right.$$

$$\left. + \int [f_7 + \int f_8 + \int f_9 + \int f_{10} (\int f_{11})] g \right\}, \quad \ell_1 = 1 \text{ or } \ell_1 = 2 \quad (5.3-2b)$$

which results in a worst-case of 2 one-dimensional integrations, 5 two-dimensional integrations, and 2 three-dimensional integrations to be performed numerically. The inner-most integrals must be evaluated to very high precision in order to evaluate the outer integrals to sufficient precision so as to reduce subtractive cancellation to acceptable levels. The result is a very slow computer program.

Under these conditions, the available computational facilities (a PDP-11/44 minicomputer) restricted the number of performance curves we were able to generate.

5.3.1 The Optimum Threshold Setting.

The optimum clipping threshold η_0 is defined as the level at which minimizes the bit error probability in the absence of jamming. This is accomplished by the first part of the computer programs for calculating the performance in partial-band noise jamming. The thresholds are normalized by the thermal noise density; thus we actually find $\eta_0/2\sigma_N^2$. The optimum thresholds found by the computer programs given in appendices H (for M=2, L=2), I (for M=4, L=2), and J (for M=2 or 4, L=3) are given in Table 5.3-1.

J. S. LEE ASSOCIATES, INC.

TABLE 5.3-1
OPTIMUM NORMALIZED CLIPPING THRESHOLD $\eta_0/2\sigma_N^2$

L	M		
	2 ($E_b/N_0 = 13.35247$ dB)	4 ($E_b/N_0 = 10.60657$ dB)	8 ($E_b/N_0 = 9.09401$ dB)
2	10.20	10.55	10.89
3	7.91	8.15	---

We note that in terms of signal power $\eta_0 = (1.89S, 1.83S, 1.79S)$ for $L=2$ and $M=(2,4,8)$; $\eta_0 = (2.19S, 2.13S)$ for $L=3$ and $M=(2,4)$. The threshold is almost a function only of S , L , and M .

J. S. LEE ASSOCIATES, INC.

5.3.2 Probability of Bit Error.

For $M=2$ and $L=2$, the computations using the program given in Appendix H were sufficiently rapid to permit obtaining a full set of curves for jamming fractions from $\gamma = 0.001$ through $\gamma = 1.0$, as shown in Figure 5.3-1.

For $M=4$ and $L=2$, the computations were much slower, due to the increased number of jamming events and the need to compute products of the function $G_L(x)$. Therefore, the program in Appendix I was used to search for the optimum value of γ for each value of E_b/N_J . To aid the speed of the search, we used the a priori knowledge that $\gamma_{opt} = 1/N$ where N is the number of hopping slots when E_b/N_J is very high, and that γ_{opt} increases as E_b/N_J decreases. Thus the computations started at $E_b/N_J = 50$ dB and decreased (in rather large steps to conserve computer time) to 0 dB. The result is the curve of $P_b(e)$ vs. E_b/N_J in worst-case partial-band noise jamming as shown for $M=4$ in Figure 5.3-2. For comparison, the envelope of the curves from Figure 5.3-1 is shown in Figure 5.3-2. We see that $M=4$ FH/RMFSK is about 2 dB better than $M=2$ FH/RMFSK in strong jamming.

Selected runs for $M=8$ and $L=2$ with $E_b/N_0 = 9.09$ dB were made in order to examine the dependence of the worst-case jamming performance upon M . These runs yielded the threshold shown in Table 5.3-1 and the following points for $\gamma=0.01$: $[E_b/N_J, P(e)] = [15, 8.963(-4)], [20, 4.680(-4)], [22.5, 2.604(-4)], [25, 1.286(-4)], [\infty, 3.91(-5)]$. From these points, a curve for $\gamma=0.01$ was constructed, and the inflection point was taken to be a point on the worst-case jamming curve. This point, estimated as the 22.5 dB point given above, is shown on Figure 5.3-2.

J. S. LEE ASSOCIATES, INC.

Although the program given in Appendix J contains code to compute $P_b(e)$ as well as $n_0/2\sigma_N^2$, excessive run time (nearly 8 hours to obtain just $n_0/2\sigma_N^2$), prevented us from allowing it to run to completion to obtain performance curves for the case L=3 hops per symbol.

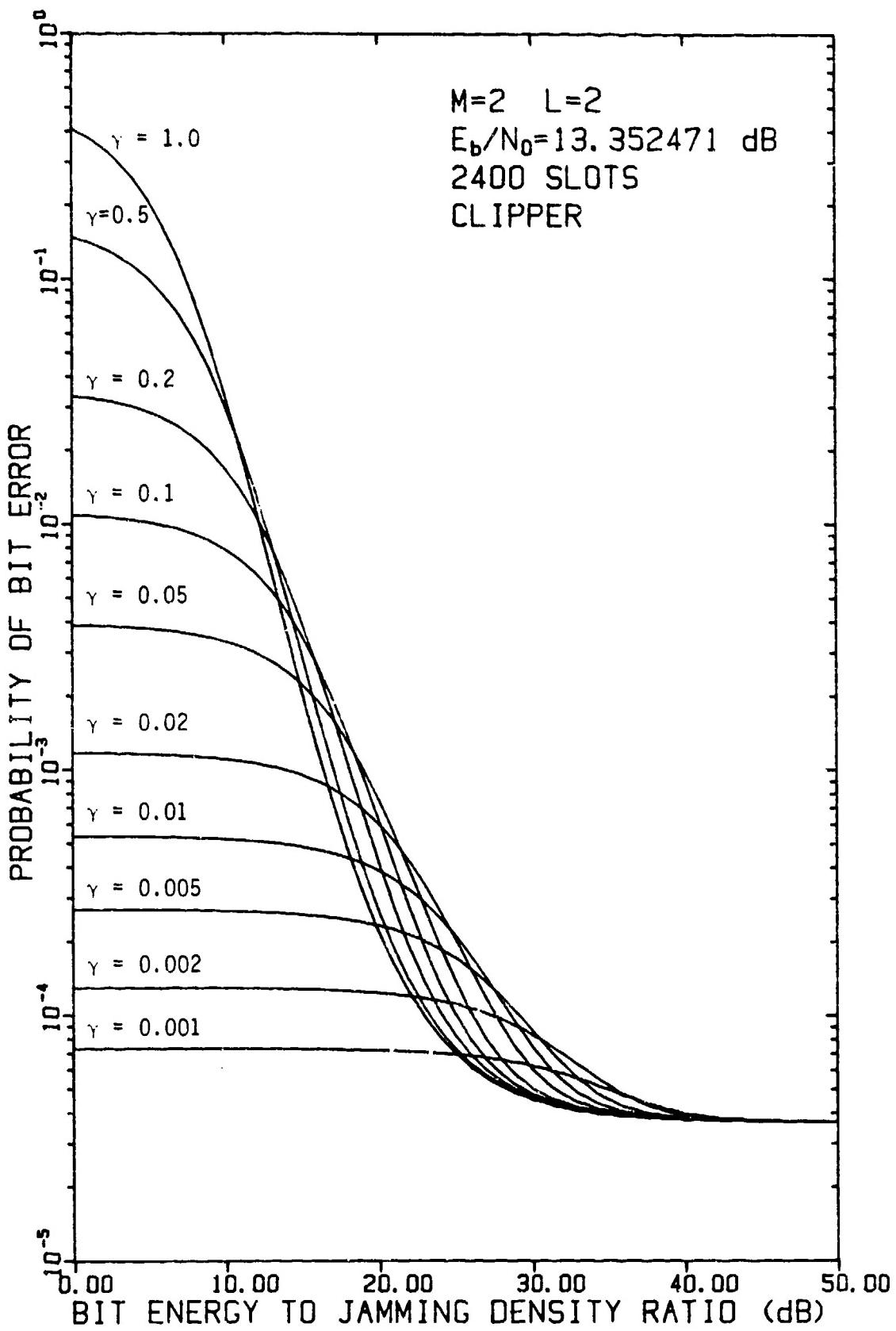


FIGURE 5.3-1 PERFORMANCE OF CLIPPER RECEIVER FOR FH/RMFSK WITH $M=2$, $L=2$ HOPS/SYMBOL, AND $E_b/N_0 = 13.35247 \text{ dB}$ (FOR $P_b(e) = 10^{-5}$ IN THE ABSENCE OF JAMMING WHEN $L=1$)

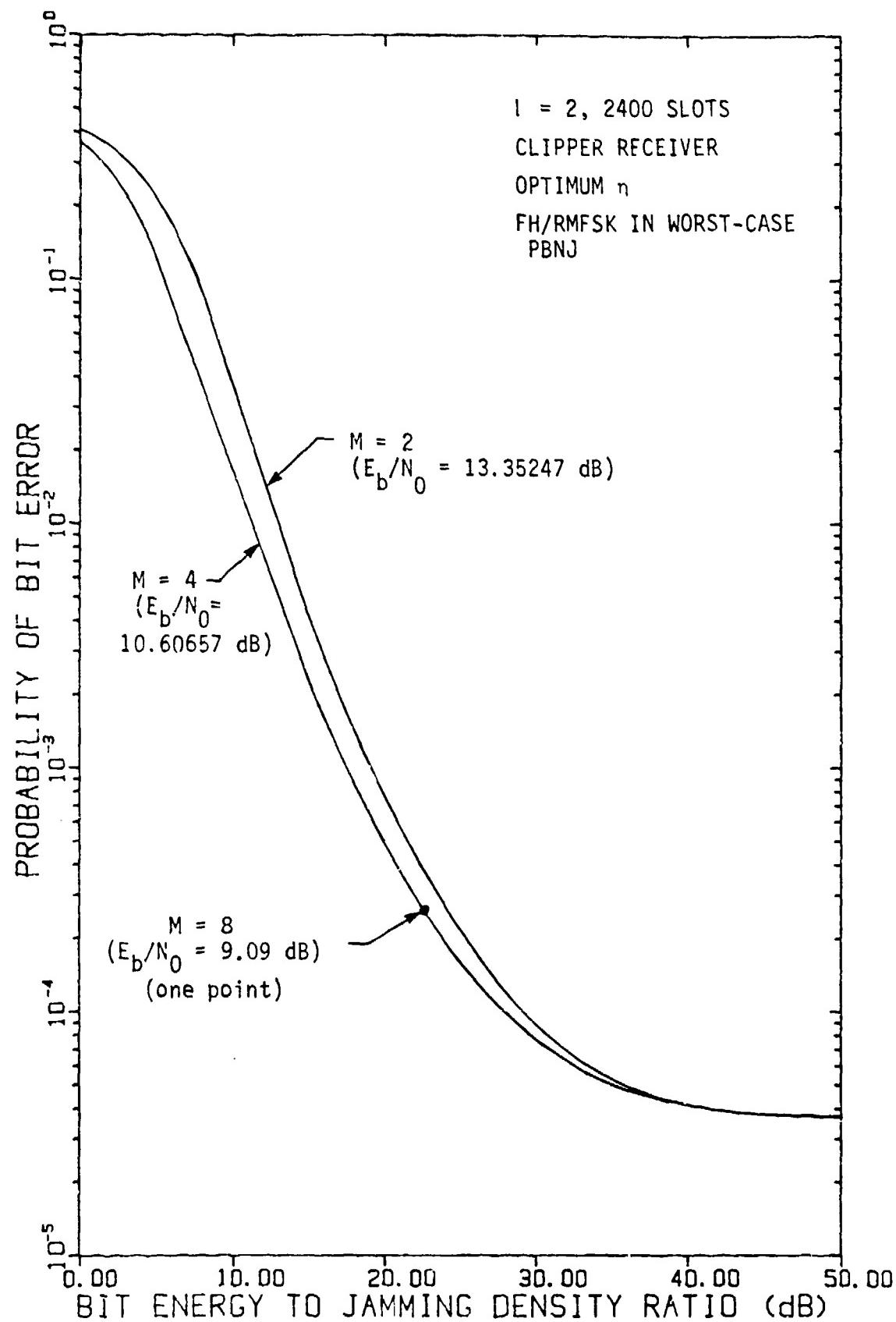


FIGURE 5.3-2 PERFORMANCE OF CLIPPER RECEIVER FOR FH/RMFSK WHEN L=2 HOPS/
 SYMBOL WITH M AS A PARAMETER AND E_b/N_0 CORRESPONDING TO
 $P_b(e) = 10^{-5}$ IN THE ABSENCE OF JAMMING (WHEN L=1 HOP/SYMBOL)

6.0 FH/RMFSK PERFORMANCE USING SQUARE-LAW SELF-NORMALIZING RECEIVER

The ECCM weighting schemes which make the (ideal) AGC and clipper soft-decision receivers for FH/MFSK and FH/RMFSK work depend upon a priori knowledge of system parameters, or else real-time measurements. (The feasibility of these measurements is discussed in a later section.) It is evident that the clipper strategy, which requires setting an SNR-dependent threshold, would be easier to implement than the AGC receiver, which requires detection of which hops are jammed and knowledge or measurement of thermal noise and jamming noise levels. Meanwhile we have seen that the hard-decision receiver accomplishes a form of ECCM protection, much in the manner of the clipper receiver - the jammed hops are prevented from dominating the decision. If there is sufficient SNR, one might well choose then to employ the hard-decision scheme, since it does not require any a priori knowledge or measurements.

In this section we consider soft-decision weighting schemes which are not predicated on using signal or noise parameters. In particular, we find the FH/RMFSK performance of a "self-normalizing" receiver in partial-band noise jamming.

6.1 THE SELF-NORMALIZATION SCHEME

The general FH/RMFSK soft-decision receiver shown in Figure 2.2-1 is rendered what we call the "self-normalizing" (SNORM) receiver by use of the weighting function

$$z_{mk} = f(x_{mk}) = \frac{x_{mk}^2}{x_{1k}^2 + x_{2k}^2 + \dots + x_{Mk}^2} . \quad (6.1-1)$$

J. S. LEE ASSOCIATES, INC.

That is, on each hop (indexed by k , $k=1,2,\dots,L$), the squared envelope samples in each channel ($m=1,2,\dots,M$) are normalized (divided) by their sum. In this manner, hops which are jammed in one or more MFSK slots can be expected to be weighted less than unjammed hops.

Thus the decision variables for the SNORM receiver are

$$\begin{aligned} z_m &= \sum_{k=1}^L z_{mk} \\ &= \sum_{k=1}^L w_k x_{mk}^2 \end{aligned} \tag{6.1-2a}$$

where

$$w_k = \left(\sum_{m=1}^M x_{mk}^2 \right)^{-1}. \tag{6.1-2b}$$

6.1.1 Single-hop Distribution of Decision Variables.

With the FH/RMFSK hopping scheme, any, none, or all of the channels can be jammed on a particular hop. Using $u_{mk} \equiv x_{mk}^2$ for the square-law envelope samples and

$$a \equiv 1/2\sigma_N^2$$

$$b \equiv 1/2\sigma_f^2, \tag{6.1-3}$$

for a general one-hop jamming event, we can write (assuming the signal is in channel 1)

$$P_{u_1}(a; c_1) = c_1 e^{-c_1 a - \rho_1} I_0(2\sqrt{\rho_1 c_1 a}), \quad a > 0; \tag{6.1-4a}$$

J. S. LEE ASSOCIATES, INC.

and

$$p_{u_m}(\alpha; c_m) = c_m e^{-c_m \alpha}, \quad m=2, \dots, M; \alpha > 0; \quad (6.1-4b)$$

where

$$c_m = \begin{cases} a, & \text{channel not jammed} \\ b, & \text{channel jammed, } m=1, 2, \dots, M. \end{cases} \quad (6.1-4c)$$

Thus the joint pdf for the square-law envelope detector samples is, conditioned on the jamming,

$$p_{\underline{u}}(\alpha_1, \dots, \alpha_M | c_1, \dots, c_M) = c_1 c_2 \dots c_M \exp \left\{ -\rho_1 - \sum_{m=1}^M c_m \alpha_m \right\} I_0(2 \sqrt{c_1 c_1 \alpha_1}), \\ \alpha_m \geq 0. \quad (6.1-5)$$

By a change of variables,

$$u_1 = \xi z_1$$

$$u_1 + u_2 = \xi(z_1 + z_2)$$

$$\sum_{i=1}^k u_i = \xi \sum_{i=1}^k z_i \quad (6.1-6)$$

$$u_1 + u_2 + \dots + u_M = \xi,$$

we can express the joint pdf of $\{z_1, z_2, \dots, z_{M-1}\}$ by, using $\underline{c} \equiv (c_1, c_2, \dots, c_M)$,

$$p_{\underline{z}}(\alpha_1, \alpha_2, \dots, \alpha_{M-1} | \underline{c}) = \int_0^\infty d\xi \xi^{M-1} p_{\underline{u}} \left[\xi \alpha_1, \xi \alpha_2, \dots, \xi \alpha_{M-1}; \xi - \sum_{m=1}^{M-1} \xi \alpha_m \right]$$

$$\begin{aligned}
 &= c_1 c_2 \dots c_M e^{-\rho_1} \int_0^\infty d\xi \xi^{M-1} \exp \left\{ -\xi \left[c_M + \sum_{m=1}^{M-1} (c_m - c_M) \alpha_m \right] \right\} \\
 &\quad \times I_0(2 \sqrt{\rho_1 c_1 \xi \alpha_1}) \\
 &= \frac{(M-1)! e^{-\rho_1} \prod_m c_m}{\left[c_M + \sum_{m=1}^{M-1} (c_m - c_M) \alpha_m \right]^M} {}_1F_1 \left[M; 1; \frac{c_1 \rho_1 \alpha_1}{c_M + \sum_{m=1}^{M-1} (c_m - c_M) \alpha_m} \right]. \tag{6.1-7}
 \end{aligned}$$

In this development we used equations 6.643.2 and 9.220.2 from [3]; ${}_1F_1(a; b, x)$ is the confluent hypergeometric function.

Note that α_M does not appear in (6.1-7). This occurrence is due to the fact that there are now only $M-1$ dependent random variables; the value of the M th channel variable is completely determined by the others. This fact can be made explicit by writing (using all variables)

$$p_z(\alpha_1, \alpha_2, \dots, \alpha_M | \underline{c}) = p_z(\alpha_1, \alpha_2, \dots, \alpha_{M-1} | \underline{c}) \delta\left(\sum_{m=1}^M \alpha_m - 1\right). \tag{6.1-8}$$

Also, we note that the domains of these variables are interdependent:

$$\begin{aligned}
 0 &\leq z_m \leq 1 \\
 0 &\leq z_i + z_j \leq 1 \quad (\text{all pairs}) \\
 0 &\leq z_i + z_j + z_k \leq 1 \quad (\text{all triples}) \\
 &\vdots \\
 0 &\leq z_1 + z_2 + \dots + z_{M-1} \leq 1 \\
 \sum_{m=1}^M z_m &\equiv 1.
 \end{aligned} \tag{6.1-9}$$

J. S. LEE ASSOCIATES, INC.

This interdependence of finite domains makes analysis and computation difficult, as will be seen below.

6.1.2 Alternate Forms.

By using the identity

$$\begin{aligned} {}_1F_1(M; 1; x) &= e^x {}_1F_1(1-M; 1; -x) \\ &= e^x \mathcal{L}_{M-1}(-x), \end{aligned} \quad (6.1-10)$$

where $\mathcal{L}_n(x)$ is the Laguerre polynomial, we realize that the joint pdf given in (6.1-7) has the form of an exponential times an $(M-1)$ -degree polynomial in $x(\underline{\alpha})$, divided by an M -degree polynomial in $y(\underline{\alpha})$:

$$p_{\underline{z}}(\underline{\alpha} | \underline{c}) = \frac{\text{const} \cdot e^{x(\underline{\alpha})} \mathcal{L}_{M-1}[-x(\underline{\alpha})]}{[y(\underline{\alpha})]^M} \quad \delta\left(\sum_m \alpha_m - 1\right) \quad (6.1-11a)$$

where

$$x(\underline{\alpha}) = c_1 \rho_1 \alpha_1 / y(\underline{\alpha}) \quad (6.1-11b)$$

$$y(\underline{\alpha}) = c_M + \sum_{m=1}^{M-1} (c_m - c_M) \alpha_m \quad (6.1-11c)$$

and

$$\text{const} = (M-1)! e^{-\rho_1} \prod_{m=1}^M c_m. \quad (6.1-11d)$$

A somewhat simpler form results from recognizing that [3, equation 8.970.1]

$$\mathcal{L}_{M-1}(x) = \frac{1}{(M-1)!} e^x \frac{d^{M-1}}{dx^{M-1}} [e^{-x} x^{M-1}]. \quad (6.1-12)$$

J. S. LEE ASSOCIATES, INC.

Applying this relation results in

$$p_z(\underline{\alpha} | \underline{c}) = \left[\frac{\prod_m c_m}{\sum_m c_m \alpha_m} \right]^M e^{-\rho_1} \frac{\partial}{\partial \rho_1}^{M-1} \left\{ \rho_1^{M-1} e^{\rho_1 \chi(\underline{\alpha})} \right\} \delta\left(\sum_m \alpha_m - 1\right). \quad (6.1-13)$$

6.1.2.1 Special case: M=2 (binary).

The various general expressions for the joint pdf reduce to the following ones for M=2:

$$p_z(\alpha_1, \alpha_2 | c_1, c_2) = \frac{c_1 c_2 e^{-\rho_1}}{(c_1 \alpha_1 + c_2 \alpha_2)^2} {}_1F_1(2;1; \frac{c_1 \rho_1 \alpha_1}{c_1 \alpha_1 + c_2 \alpha_2}) \delta(\alpha_1 + \alpha_2 - 1) \quad (6.1-14a)$$

$$= \frac{c_1 c_2 e^{-\rho_1}}{[c_2 + (c_1 - c_2) \alpha_1]^2} \exp \left\{ \frac{c_1 \rho_1 \alpha_1}{c_2 + (c_1 - c_2) \alpha_1} \right\} \left[1 + \frac{c_1 \rho_1 \alpha_1}{c_2 + (c_1 - c_2) \alpha_1} \right] \delta(\alpha_1 + \alpha_2 - 1) \quad (6.1-14b)$$

$$= \frac{c_1 c_2 e^{-\rho_1}}{[c_2 + (c_1 - c_2) \alpha_1]^2} \frac{\partial}{\partial \rho_1} \left[\rho_1 \exp \left\{ \frac{c_1 \rho_1 \alpha_1}{c_2 + (c_1 - c_2) \alpha_1} \right\} \right] \delta(\alpha_1 + \alpha_2 - 1) \quad (6.1-14c)$$

with

$$0 \leq \alpha_1 \leq 1, \quad 0 \leq \alpha_2 = 1 - \alpha_1 \leq 1. \quad (6.1-14d)$$

6.1.2.2 Special case: no jamming.

For no jamming, $c_1 = c_2 = \dots = c_M$ (also true for all channels jammed), the general pdf reduces to

$$p_z(\underline{\alpha} | c_1 = c_2 = \dots = c_M) = (M-1)! e^{-\rho_1} {}_1F_1(M;1;\rho_1 \alpha_1) \delta\left(\sum_m \alpha_m - 1\right) \quad (6.1-15a)$$

$$= (M-1)! e^{-\rho_1 + \rho_1 \alpha_1} \mathcal{L}_{M-1}(-\rho_1 \alpha_1) \delta\left(\sum_m \alpha_m - 1\right) \quad (6.1-15b)$$

$$= e^{-\rho_1} \frac{\partial^{M-1}}{\partial \rho_1^{M-1}} \left\{ \rho_1^{M-1} e^{\rho_1 \alpha_1} \right\} \delta\left(\sum_m \alpha_m - 1\right). \quad (6.1-15c)$$

6.1.3 Conditional pdf's for M=2.

We now show the explicit expressions for the decision variable for the binary case; there is only one decision variable $z \equiv z_1$ since $z_2 = 1 - z_1$.

6.1.3.1 Single hop/bit case ($L=1$).

As far as computation of error probabilities is concerned, the $L=1$ case of the SNORM receiver pdf's is not needed since the normalization does not affect the outcome of the decision; we know in advance that the result will be the same as if no normalization were employed. However, to go on to the $L=2$ case, we need the $L=1$ pdf's.

Using $K \triangleq \frac{\sigma^2_T}{\sigma^2_N}$ as in previous analyses, the pdf's conditioned on the possible jamming events $\underline{v} = (v_1, v_2)$ are as follows:

$$p_1[z | \underline{v} = (0,0)] = e^{-\rho_N} {}_1F_1(2;1;\rho_N z) \quad (6.1-16a)$$

$$= e^{-\rho_N + \rho_N z} (1 + \rho_N z). \quad (6.1-16b)$$

$$p_1[z | \underline{v} = (0,1)] = \frac{Ke^{-\rho_N}}{[1 + (K-1)z]^2} {}_1F_1\left(2;1; \frac{K\rho_N z}{1 + (K-1)z}\right) \quad (6.1-17a)$$

$$= \frac{Ke^{-\rho_N}}{[1 + (K-1)z]^2} \exp\left[\frac{K\rho_N z}{1 + (K-1)z}\right] \left(1 + \frac{K\rho_N z}{1 + (K-1)z}\right) \quad (6.1-17b)$$

$$p_1[z | \underline{v} = (1,0)] = \frac{Ke^{-\rho_T}}{[K - (K-1)z]^2} {}_1F_1\left(2;1; \frac{\rho_T z}{K - (K-1)z}\right) \quad (6.1-18a)$$

$$= \frac{Ke^{-\rho_T}}{[K - (K-1)z]^2} \exp\left[\frac{\rho_T z}{K - (K-1)z}\right] \left(1 + \frac{\rho_T z}{K - (K-1)z}\right) \quad (6.1-18b)$$

$$p_1[z | \underline{v} = (1,1)] = e^{-\rho_T} {}_1F_1(2;1;\rho_T z) \quad (6.1-19a)$$

$$= e^{-\rho_T + \rho_T z} (1 + \rho_T z). \quad (6.1-19b)$$

J. S. LEE ASSOCIATES, INC.

For all of these expressions, the domain of z is $0 \leq z \leq 1$. Note that when there is no signal ($\rho_N = \rho_T = 0$), the variable z is uniformly distributed when both channels have the same noise power.

6.1.3.2 Two hops/bit case ($L=2$).

To obtain the pdf for $L=2$, it is necessary to convolve the expressions (6.1-16) to (6.1-19) with each other for the jamming events $\underline{\ell} = \underline{v}_1 + \underline{v}_2$. The general form of the convolution is

$$p_2(z|\underline{\ell} = \underline{v}_1 + \underline{v}_2) = \int_{\max(0, z-1)}^{\min(1, z)} dv p_1(z-v|\underline{v}_1) p_1(v|\underline{v}_2), \quad 0 \leq z \leq 2. \quad (6.1-2)$$

There are ten distinguishable jamming events, two for $\underline{\ell} = (1,1)$ and one each for other $\underline{\ell}$. For three of these ten cases, the convolution shown in (6.1-20) can be performed analytically without too much difficulty; the cases are the ones in which both channels are jammed or not jammed on a given hop: $\underline{\ell} = (0,0)$, $(1,1)^*$, $(2,2)$. For these cases, the pdf for one hop can be written

$$p_1(z) = e^{-\rho_1} \frac{\partial}{\partial \rho_1} \rho_1 e^{\rho_1 z}, \quad 0 \leq z \leq 1 \quad (6.1-2)$$

The convolution then takes the form

$$\begin{aligned} p_2(z|\rho_1, \rho_2) &= e^{-\rho_1 - \rho_2} \frac{\partial^2}{\partial \rho_1 \partial \rho_2} \rho_1 \rho_2 \int_{\max(0, z-1)}^{\min(1, z)} dv e^{\rho_1(z-v) + \rho_2 v} \\ &= e^{-\rho_1 - \rho_2} \frac{\partial^2}{\partial \rho_1 \partial \rho_2} \frac{\rho_2 \rho_1}{\rho_2 + \rho_1} \cdot \begin{cases} e^{\rho_2 z - \rho_1 z}, & 0 \leq z \leq 1 \\ e^{\rho_2 + \rho_1(z-1)} - e^{\rho_1 + \rho_2(z-1)}, & 1 \leq z \leq 2. \end{cases} \end{aligned} \quad (6.1-2)$$

*i.e., the case of $\underline{\ell} = (1,1)$ where $\underline{v}_1 = (1,1)$ and $\underline{v}_2 = (0,0)$ or vice versa.

J. S. LEE ASSOCIATES, INC.

Carrying out the partial differentiations results in

$$p_2(z|\rho_1, \rho_2) = \frac{e^{-\rho_1-\rho_2}}{(\rho_1-\rho_2)^3} \left\{ e^{\rho_1 z} \left[-2\rho_1\rho_2 + \rho_1^2(\rho_1-\rho_2)z \right] \right. \quad (6.1-23a)$$

$$\left. + e^{\rho_2 z} \left[2\rho_1\rho_2 + \rho_2^2(\rho_1-\rho_2)z \right] \right\}, \quad \rho_1 \neq \rho_2, 0 \leq z \leq 1; \\ = e^{-2\rho+\rho z} z(1 + \rho z + \rho^2 z^2/6), \quad \rho_1 = \rho_2 = \rho, \quad 0 < z < 1; \quad (6.1-23b)$$

$$= \frac{1}{(\rho_1-\rho_2)^3} \left\{ e^{\rho_2(z-2)} \left[-2\rho_1\rho_2 + \rho_1^2(\rho_1-\rho_2) + \right. \right. \\ \left. \left. \rho_2(\rho_1-\rho_2)(\rho_1^2-\rho_1\rho_2-\rho_2)(z-1) \right] \right\}$$

$$+ e^{\rho_1(z-2)} \left[2\rho_1\rho_2 + \rho_2^2(\rho_1-\rho_2) \right. \\ \left. + \rho_1(\rho_1-\rho_2)(\rho_2^2-\rho_1\rho_2-\rho_1)(z-1) \right] \quad (6.1-23c)$$

$$= e^{-2\rho+\rho z} \left[1 + \rho + \rho^2/6 - (1-\rho^2/2)(z-1) - \rho(1+\rho/2)(z-1)^2 \right. \\ \left. - (\rho^2/6)(z-1)^3 \right], \quad \rho_1 = \rho_2 = \rho, \quad 1 < z \leq 2. \quad (6.1-23d)$$

This expression is applied to the pertinent jamming events using the following table:

k_1	ρ_1	ρ_2	
0	ρ_N	ρ_N	
1	ρ_N	ρ_T	
2	ρ_T	ρ_T	(6.1-24)

J. S. LEE ASSOCIATES, INC.

The other seven cases must be handled by numerical convolution.
(We have found an analytical expression, but it is no easier to compute than the convolutions.)

6.1.4 Conditional pdf's for M=4.

The system analysis for $M > 2$ becomes very difficult, as we now demonstrate for $M=4$.

6.1.4.1 Single hop/symbol case ($L=1$).

For $M=4$ and $L=1$ there are sixteen possible jamming events, described by the vector $\underline{v} = (v_1, v_2, v_3, v_4)$, where $v_m = 1$ if the m th symbol frequency slot is jammed, and $v_m = 0$ if not. These events give rise to the conditional pdf

$$p(\underline{z}|\underline{v}) = \frac{6\mu_1}{[y(\underline{z})]^4} e^{-\rho_1 + \mu_2 \rho_1 z_1/y(\underline{z})} \mathcal{L}_3[-\mu_2 \rho_1 z_1/y(\underline{z})], \quad (6.1-25a)$$

where

$$\mathcal{L}_3(-u) = 1 + 3u + \frac{3}{2}u^2 + \frac{1}{6}u^3 \quad (6.1-25b)$$

and the parameters μ_1, μ_2 , and ρ_1 and the polynomials $y(\underline{z})$ are listed in Table 6.1-1. Since $z_4 \equiv 1 - z_1 - z_2 - z_3$, it does not appear in the pdf. We note that z_1 always appears in the conditional pdf, while z_2 and z_3 may or may not appear.

It is understood that the domain of values for the variables is $(z_1, z_2, z_3) \in \Omega_{4,1}$, where $\Omega_{4,1}$ is the volume

$$\Omega_{4,1} : \begin{cases} 0 \leq z_i \leq 1, i = 1, 2, 3; \\ 0 \leq z_i + z_j \leq 1, \text{ all pairs}; \\ 0 \leq z_1 + z_2 + z_3 \leq 1. \end{cases} \quad (6.1-26)$$

J. S. LEE ASSOCIATES, INC.

TABLE 6.1-1

L=1 PROBABILITY DENSITIES FOR M=4

	$v_1 v_2 v_3 v_4$	ρ_1	μ_1	μ_2	$y(z)$
A	0 0 0 0		1	1	1
B	0 0 0 1		K^3		$1 + (K-1)(z_1+z_2+z_3)$
C	0 0 1 0		K^3		$K - (K-1)z_3$
D	0 0 1 1	ρ_N	K^2		$1 + (K-1)(z_1+z_2)$
E	0 1 0 0		K^3	K	$K - (K-1)z_2$
F	0 1 0 1		K^2		$1 + (K-1)(z_1+z_3)$
G	0 1 1 0		K^2		$K - (K-1)(z_2+z_3)$
H	0 1 1 1		K		$1 + (K-1)z_1$
I	1 0 0 0		K^3		$K - (K-1)z_1$
J	1 0 0 1		K^2		$1 + (K-1)(z_2+z_3)$
K	1 0 1 0		K^2		$K - (K-1)(z_1+z_3)$
L	1 0 1 1		K		$1 + (K-1)z_2$
M	1 1 0 0	ρ_T	K^2	1	$K - (K-1)(z_1+z_2)$
N	1 1 0 1		K		$1 + (K-1)z_3$
O	1 1 1 0		K		$K - (K-1)(z_1+z_2+z_3)$
P	1 1 1 1		1		1

Form:

$$p_1(z|v) = \frac{6\mu_1}{[y(z)]^4} e^{-\rho_1 + \mu_2 \rho_1 z_1/y(z)} \mathcal{L}_3(-\mu_2 \rho_1 z_1/y(z))$$

$$\mathcal{L}_3(u) = 1 + 3u + \frac{3}{2}u^2 + \frac{1}{6}u^3$$

$$\underline{z} = (z_1, z_2, z_3) \in \Omega_{4,1}$$

$$z_4 = 1 - z_1 - z_2 - z_3$$

J. S. LEE ASSOCIATES, INC.

The domain $\Omega_{4,1}$ may also be described as the volume included by the planes $z_1=0$, $z_2=0$, $z_3=0$, and $z_1+z_2+z_3=1$, as illustrated by Figure 6.1-1. It is obvious from the mutual constraints among the variables that they are statistically dependent.

6.1.4.2 Two hop/symbol case ($L=2$) .

Since the four SNORM variables are dependent, we cannot analyze the $M=4$, $L=2$ case by finding the two-sample distributions of the separate channels as we did for other receivers. The convolution must be done in three dimensions ($M-1$ dimensions for the general case). The concept for doing this is unusual, but can be visualized. Figure 6.1-2 illustrates the fact that multi-dimensional convolution of two pdf's involves integration over the volume which is the intersection of the domains of the pdf's. In Figure 6.1-2(b), the simple case when the point (z_1, z_2, z_3) lies inside the domain of $p_1(z)$ is shown; this yields a rectangular-sided volume. If (z_1, z_2, z_3) lies outside the domain of $p_1(z)$, the intersection is much more complicated.

By careful study we have determined that the pdf for the SNORM receiver's decision variables for $M=4$ and $L=2$ has the general form

$$p_2(z | \underline{z} = \underline{v}_1 + \underline{v}_2) = \int_{A_1}^{B_1} dv_1 \int_{A_2}^{B_2} dv_2 \int_{A_3}^{B_3} dv_3 p_1(\underline{v} | \underline{v}_1) p_1(z - \underline{v} | \underline{v}_2), \quad (6.1-27a)$$

where

$$A_1 = \max(0, z_1 - 1)$$

$$B_1 = \min(1, z_1)$$

$$A_2 = \max(0, z_1 + z_2 - v_1 - 1) \quad (6.1-27b)$$

$$B_2 = \min(1 - v_1, z_2)$$

$$A_3 = \max(0, z_1 + z_2 + z_3 - v_1 - v_2 - 1)$$

$$B_3 = \min(1 - v_1 - v_2, z_3).$$

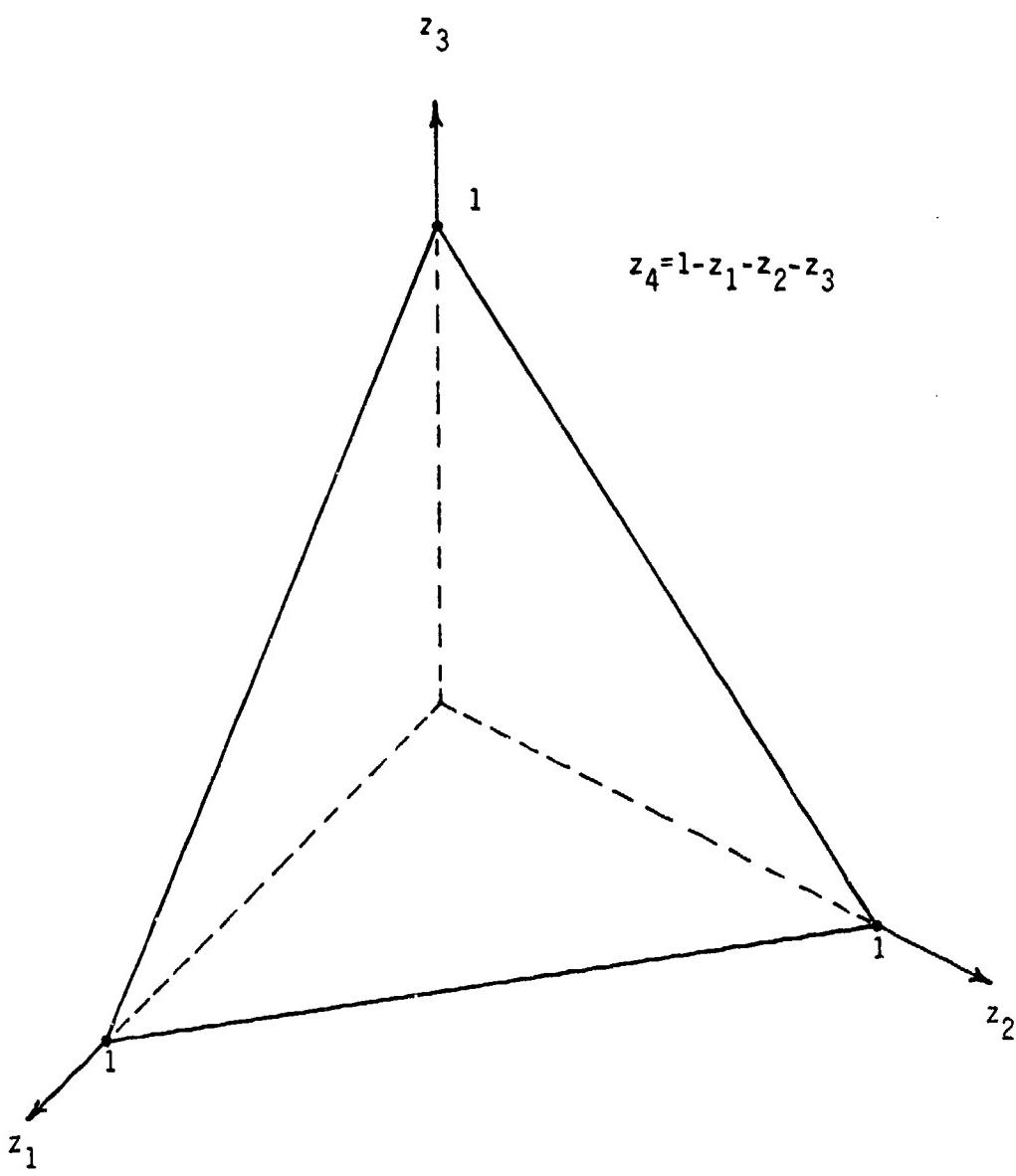
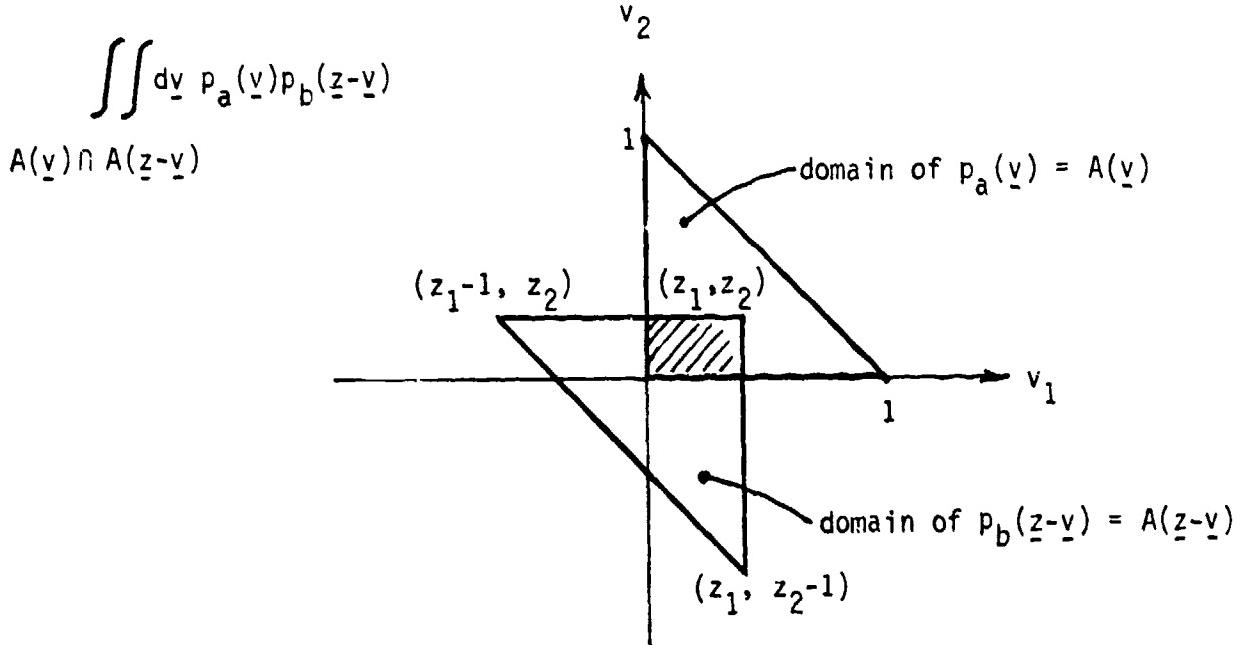
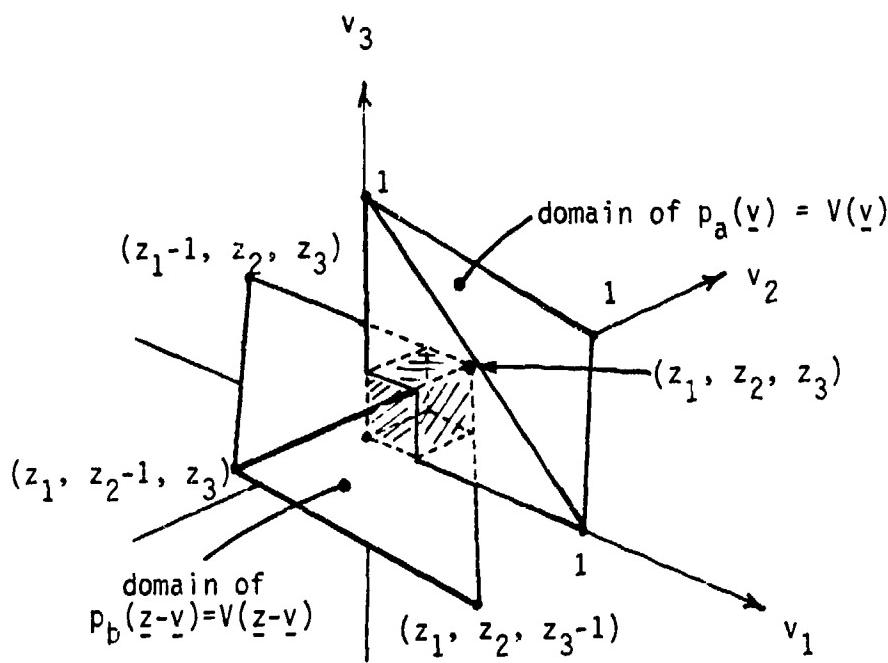


FIGURE 6.1-1 REGION CONTAINING DECISION VARIABLES



(a) two-dimensional convolution



(b) three-dimensional convolution

FIGURE 6.1-2 TWO- AND THREE-DIMENSIONAL CONVOLUTIONS

Now, since there are sixteen cases of $p_1(z|\underline{v})$ for $L=1$, there are $(16)^2 = 256$ cases for $L=2$. However since the numbering of symbol channels is arbitrary, there can be considered to be fewer, distinguishable jamming events. These are fully enumerated in Section 6.2. What we wish to note here is that if neither of the densities in (6.1-27) contains v_3 , the integral can be simplified to

$$p_2(z|\underline{z}) = \int_{A_1}^{B_1} dv_1 \int_{A_2}^{B_2} dv_2 p_1(v|\underline{v}_1) p_1(z-v|\underline{v}_2) (B_3 - A_3) \quad (6.1-28a)$$

where

$$\begin{aligned} B_3 - A_3 &= \min(1-v_1-v_2, v_3) - \max(0, z_1+z_2+z_3-v_1-v_2-1) \\ &= \frac{1}{2} \left\{ 2-z_1-z_2 - |1-v_1-v_2-z_3| - |z_1+z_2+z_3-v_1-v_2-1| \right\}. \end{aligned} \quad (6.1-28b)$$

If neither pdf in (6.1-27) contains v_2 or v_3 , the integral can be further simplified to

$$p_2(z|\underline{z}) = \int_{A_1}^{B_1} dv_1 p_1(v_1|\underline{v}_1) p_1(z_1-v_1|\underline{v}_2) \int_{A_2}^{B_2} dv_2 (B_3 - A_3), \quad (6.1-29)$$

where

$$\begin{aligned} \int_{A_2}^{B_2} dv_2 (B_3 - A_3) &= \frac{1}{2} (2-z_1-z_2) (B_2 - A_2) \\ &\quad - \frac{1}{4} \left\{ (B_2+v_1+z_3-1) |B_2+v_1+z_3-1| \right. \\ &\quad \left. - (A_2+v_1+z_3-1) |A_2+v_1+z_3-1| \right. \\ &\quad + (B_2+v_1+1-z_1-z_2-z_3) |B_2+v_1+1-z_1-z_2-z_3| \\ &\quad \left. - (A_2+v_1+1-z_1-z_2-z_3) |A_2+v_1+1-z_1-z_2-z_3| \right\}, \end{aligned} \quad (6.1-30)$$

J. S. LEE ASSOCIATES, INC.

since

$$\int_A^B dx |x-a| = \int_A^B dx |a-x| = \frac{1}{2} (B-a)|B-a| - \frac{1}{2} (A-a)|A-a|. \quad (6.1-3)$$

Now, if v_3 is in the integrand of (6.1-27) but v_2 is not, we simply "switch labels" on v_2 and v_3 to get (6.1-28) with v_3 replacing v_2 .

6.2 JAMMING EVENTS AND ERROR PROBABILITY FOR L=2

We now extend the conditional distribution analysis in the last section to obtain the BER for the FH/RMFSK SNORM receiver under partial-band noise jamming. Since the jammed error for L=1 is the same for other receivers, we proceed to the case of L=2.

6.2.1 Jamming Events and Probabilities for M=2.

For L=2 and M=2 there are $2^{ML}=16$ elementary jamming events. As mentioned previously, for the SNORM receiver, only ten of these events are distinguishable in terms of jamming effects. These are listed in Table 6.2-1, along with the single-hop events which produce them and the probabilities of the L=2 events.

The error event, assuming the signal is in channel 1, is $z_1 < z_2 \equiv 2-z_1$. Thus

$$P_b(e; \gamma) = Pr\{z_1 < 1\} = \sum_{\underline{x}} Pr\{\underline{x}\} Pr\{z_1 < 1 | \underline{x}\}. \quad (6.2-1)$$

The conditional error probabilities in (6.2-1) are calculated by integrating the pdf's shown previously in equation (6.1-23) or the convolution

J. S. LEE ASSOCIATES, INC.

TABLE 6.2-1

JAMMING EVENTS AND PROBABILITIES FOR M=2, L=2

<u>l</u>	<u>v</u> ₁	<u>v</u> ₂	# events	Probability
0,0	0,0	0,0	1	π_0^2
0,1	0,1	0,0	2	$2\pi_0\pi_1$
0,2	0,1	0,1	1	π_1^2
1,0	1,0	0,0	2	$2\pi_0\pi_1$
1,1	0,0	1,1	2	$2\pi_0\pi_2$
	0,1	1,0	2	$2\pi_1^2$
1,2	0,1	1,1	2	$2\pi_1\pi_2$
2,0	1,0	1,0	1	π_1^2
2,1	1,0	1,1	2	$2\pi_1\pi_2$
2,2	1,1	1,1	1	π_2^2
Totals:		16	1	

$$\pi_0 = \frac{\binom{N-2}{q}}{\binom{N}{q}} = \frac{(N-q)(N-q-1)}{N(N-1)}, \quad \pi_1 = \frac{\binom{N-2}{q-1}}{\binom{N}{q}} = \frac{q(N-q)}{N(N-1)}, \quad \pi_2 = \frac{\binom{N-2}{q-2}}{\binom{N}{q}} = \frac{q(q-1)}{N(N-1)}$$

$$p_2(z | l=v_2+v_2) = \int_{\max(0, z-1)}^{\min(1, z)} dv p_1(v | v_1) p_1(z-v | v_2)$$

J. S. LEE ASSOCIATES, INC.

of the pdf's in equations (6.1-16) to (6.1-19). The result is

$$\begin{aligned}
 P_b(e; \gamma) = & \pi_0^2 \cdot \frac{1}{2} e^{-\rho_N} (1+\rho_N/3) \\
 & + 2\pi_0\pi_1 \int_0^1 dv (1-v)e^{-\rho_N v} p_1(v|0,1) \\
 & + \pi_1^2 \int_0^1 dv p_1(v|0,1) \cdot \frac{K(1-v)}{v+K(1-v)} \exp \left\{ \frac{-\rho_N v}{v+K(1-v)} \right\} \\
 & + 2\pi_0\pi_1 \int_0^1 dv (1-v)e^{-\rho_N v} p_1(v|1,0) \\
 & + 2\pi_0\pi_2 \frac{1}{(\rho_N - \rho_T)^3} \left\{ e^{-\rho_T} \left[\rho_N(\rho_N - \rho_T) - (\rho_N + \rho_T) \right] \right. \\
 & \quad \left. + e^{-\rho_N} \left[\rho_T(\rho_N - \rho_T) + (\rho_N + \rho_T) \right] \right\} \\
 & + 2\pi_1^2 \int_0^1 dv p_1(v|0,1) \cdot \frac{1-v}{Kv+1-v} \exp \left\{ \frac{-K\rho_T v}{Kv+1-v} \right\} \\
 & + 2\pi_1\pi_2 \int_0^1 dv (1-v)e^{-\rho_T v} p_1(v|0,1) \\
 & + \pi_1^2 \int_0^1 dv p_1(v|1,0) \cdot \frac{1-v}{Kv+1-v} \exp \left\{ \frac{-K\rho_T v}{Kv+1-v} \right\} \\
 & + 2\pi_1\pi_2 \int_0^1 dv (1-v)e^{-\rho_T v} p_1(v|1,0) \\
 & + \pi_2^2 \cdot \frac{1}{2} e^{-\rho_T} (1+\rho_T/3). \tag{6.2-2}
 \end{aligned}$$

J. S. LEE ASSOCIATES, INC.

In this expression we have used the fact that

$$\begin{aligned}
 & \int_0^1 dz \int_{\max(0, z-1)}^{\min(1, z)} dv \ p_1(v | \underline{v}_1) \ p_1(z-v | \underline{v}_2) \\
 &= \int_0^1 dz \int_0^z dv \ p_1(v | \underline{v}_1) \ p_1(z-v | \underline{v}_2) \\
 &= \int_0^1 dv \ p_1(v | \underline{v}_1) \int_0^{1-v} dz \ p_1(z | \underline{v}_2). \tag{6.2-3}
 \end{aligned}$$

Also, the parameters K , ρ_N , and ρ_T are

$$K = \frac{\sigma_T^2}{\sigma_N^2}, \quad \rho_N = \frac{1}{2} \cdot \frac{E_b}{N_0}, \quad \rho_T = \frac{1}{2} \cdot \frac{E_b}{N_T}. \tag{6.2-4}$$

6.2.2 Jamming Events and Probabilities for M=4.

For $L=2$ and $M=4$, there are 256 elementary jamming events, which can be represented by 47 distinguishable events. These are listed in Table 6.2-2, along with their probabilities of occurrence. The joint pdf of the decision variables, given the representative jamming event shown in the table, is the convolution (6.1-27) with the single-hop pdf's selected from Table 6.1-1 as indicated.

The error event for $M=4$ and $L=2$ is the complement of the condition for a correct symbol decision, so that the conditional error probability is

$$\begin{aligned}
 P_s(e | \underline{z}) &= 1 - \Pr\{z_1 > z_2, z_1 > z_3, z_1 > z_4 = 2 - z_1 - z_2 - z_3 | \underline{z}\} \\
 &= 1 - \Pr\{z_1 > z_2, z_1 > z_3, 2z_1 > 2 - z_2 - z_3 | \underline{z}\} \tag{6.2-5a}
 \end{aligned}$$

$$\equiv 1 - P_s(C | \underline{z}). \tag{6.2-5b}$$

J. S. LEE ASSOCIATES, INC.

TABLE 6.2-2
JAMMING EVENTS AND PROBABILITIES FOR M=4, L=2

<u>L</u>	cases*	# events	prob.	<u>L</u>	cases*	# events	prob.
0 0 0 0	A+A	1	π_0^2	1 2 2 0	F+N	6	$6\pi_2\pi_3$
0 1 0 0	A+E	6	$6\pi_0\pi_1$	1 2 2 1	H+N	6	$6\pi_3^2$
0 1 1 0	A+G	6	$6\pi_0\pi_2$		G+P	6	$6\pi_2\pi_4$
	C+E	6	$6\pi_1^2$	1 2 2 2	H+P	2	$2\pi_3\pi_4$
0 1 1 1	A+H	2	$2\pi_0\pi_3$				
	C+F	6	$6\pi_1\pi_2$	2 0 0 0	I+I	1	π_1^2
0 2 0 0	E+E	3	$3\pi_1^2$	2 1 0 0	I+M	6	$6\pi_1\pi_2$
0 2 1 0	E+G	12	$12\pi_1\pi_2$	2 1 1 0	I+N	6	$6\pi_1\pi_3$
0 2 1 1	E+H	6	$6\pi_1\pi_3$		K+M	6	$6\pi_2^2$
	F+G	6	$6\pi_2^2$	2 1 1 1	I+P	2	$2\pi_1\pi_4$
0 2 2 0	G+G	3	$3\pi_2^2$		K+N	6	$6\pi_2\pi_3$
0 2 2 1	G+H	6	$6\pi_2\pi_3$	2 2 0 0	M+M	3	$3\pi_2^2$
0 2 2 0	H+H	1	π_3^2	2 2 1 0	M+N	12	$12\pi_2\pi_3$
				2 2 1 1	M+P	6	$6\pi_2\pi_4$
1 0 0 0	A+I	2	$2\pi_0\pi_1$		L+N	6	$6\pi_3^2$
1 1 0 0	A+M	6	$6\pi_0\pi_2$	2 2 2 0	N+N	3	$3\pi_3^2$
	E+I	6	$6\pi_1^2$	2 2 2 1	N+P	6	$6\pi_3\pi_4$
1 1 1 0	A+N	6	$6\pi_0\pi_3$	2 2 2 2	P+P	1	π_4^2
	G+I	6	$6\pi_1\pi_2$		Totals:	256	1
	C+M	12	$12\pi_1\pi_2$				
1 1 1 1	A+P	2	$2\pi_0\pi_4$		*cases (A-P): See Table 6.1-1		
	H+I	2	$2\pi_1\pi_3$				
	C+N	6	$6\pi_1\pi_3$				
	F+K	6	$6\pi_2^2$				
1 2 0 0	E+M	6	$6\pi_1\pi_2$				
1 2 1 0	E+N	12	$12\pi_1\pi_3$				
	G+M	12	$12\pi_2^2$				
1 2 1 1	G+N	12	$12\pi_1\pi_3$				
	E+P	6	$6\pi_1\pi_4$				
	H+M	6	$6\pi_2\pi_3$				

$$\pi_0 = \frac{(N-q)(N-q-1)(N-q-2)(N-q-3)}{N(N-1)(N-2)(N-3)}$$

$$\pi_1 = \frac{q(N-q)(N-q-1)(N-q-2)}{N(N-1)(N-2)(N-3)}$$

$$\pi_2 = \frac{q(q-1)(N-q)(N-q-1)}{N(N-1)(N-2)(N-3)}$$

$$\pi_3 = \frac{q(q-1)(q-2)(N-q)}{N(N-1)(N-2)(N-3)}$$

$$\pi_4 = \frac{q(q-1)(q-2)(q-3)}{N(N-1)(N-2)(N-3)}$$

J. S. LEE ASSOCIATES, INC.

From this expression we observe that the probability of a correct symbol decision is obtained from the joint pdf of (z_1, z_2, z_3) by integrating it over the volume Ω_C implied in (6.2-5):

$$P_s(C|\underline{z}) = \iiint_{\Omega_C} dz_1 dz_2 dz_3 p_2(\underline{z} | \underline{z}). \quad (6.2-6)$$

As illustrated in Figure 6.2-1, the volume Ω_C may be described as that enclosed by the planes $z_2=0$, $z_3=0$, $2z_1+z_2+z_3=2$, $z_1+z_2+z_3=2$, $z_1=z_2$, and $z_1=z_3$. Thus

$$P_s(C|\underline{z}) = \int_{A_4}^{B_4} dz_1 \int_{A_5}^{B_5} dz_2 \int_{A_6}^{B_6} dz_3 p_2(\underline{z} | \underline{z}), \quad (6.2-7a)$$

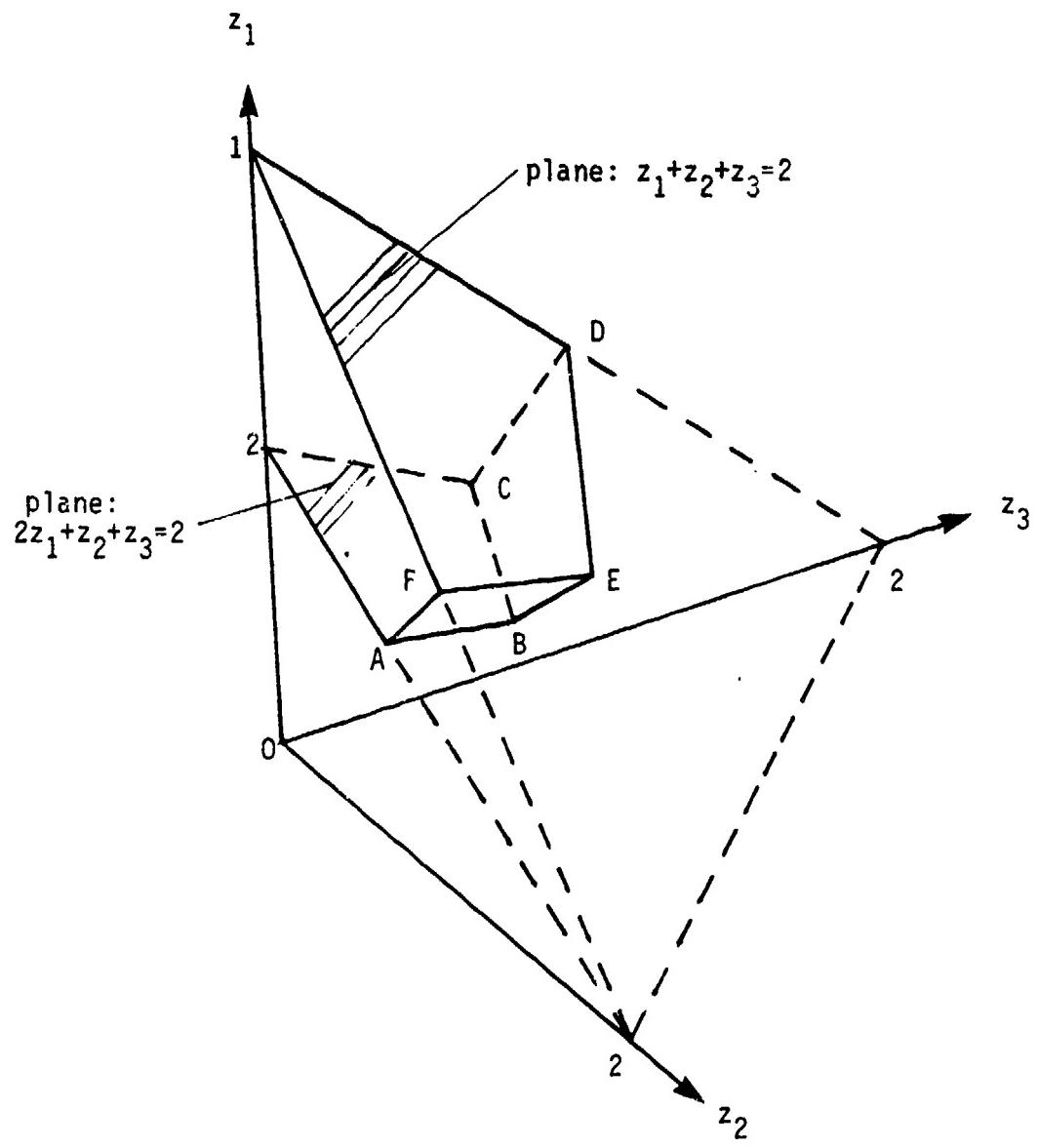
where $A_4 = 1/2$, $B_4 = 2$

$$A_5 = \max(0, 2-3z_1)$$

$$B_5 = \min(z_1, 2-z_1) \quad (6.2-7b)$$

$$A_6 = \max(0, 2-2z_1-z_2)$$

$$B_6 = \min(z_1, 2-z_1-z_2).$$



points: (z_1, z_2, z_3)

A: $(2/3, 2/3, 0)$ D: $(1, 0, 1)$

B: $(1/2, 1/2, 1/2)$ E: $(2/3, 2/3, 2/3)$

C: $(2/3, 0, 2/3)$ F: $(1, 1, 0)$

FIGURE 6.2-1 VOLUME OF INTEGRATION FOR CORRECT SYMBOL DECISION, M=4, L=2

J. S. LEE ASSOCIATES, INC.

6.3 AN ALTERNATE APPROACH FOR M=2 AND L=3 HOPS/SYMBOL

In order to obtain a more computationally tractable form for the performance of the self-normalizing receiver in partial-band noise jamming, we may proceed as follows. The probability of a symbol error is

$$\Pr(e) = \Pr\{z < 3/2\}$$

$$\begin{aligned} &= E_{\underline{v}}\{\Pr\{z < 3/2 | \underline{v}\}\} \\ &= E_{\underline{v}}\left\{\int_0^{3/2} p_3(z | \underline{v}) dz\right\} \end{aligned} \quad (6.3-1)$$

where $p_3(z | \underline{v})$ is the probability density function conditioned on jamming event \underline{v} .

If we interchange the order of integration with respect to z and expectation with respect to \underline{v} in (6.3-1), we obtain

$$\Pr(e) = \int_0^{3/2} E_{\underline{v}}\{p_3(z | \underline{v})\} dz. \quad (6.3-2)$$

The expectation in (6.3-2) may be written as

$$\begin{aligned} p_3(z) &\triangleq E_{\underline{v}}\{p_3(z | \underline{v})\} \\ &= E_{\underline{v}}\{p_1(z | \underline{v}) * p_2(z | \underline{v})\} \\ &= p_1(z) * p_2(z) \end{aligned} \quad (6.3-3)$$

where the operator $*$ denotes convolution and

$$p_1(z) \triangleq E_{\underline{v}_1}\{p_1(z | \underline{v}_1)\} \quad (6.3-4)$$

with \underline{v}_1 being the first column of the event matrix \underline{v} and

$$p_2(\zeta) \triangleq E_{\underline{v}_2} \{ p_2(\zeta | \underline{v}_2) \} \quad (6.3-5)$$

with \underline{v}_2 being the last two columns of the matrix \underline{v} . The expectation in (6.3-4) is given by

$$p_1(\zeta) = \pi_0 p_{00}(\zeta) + \pi_1 p_{01}(\zeta) + \pi_1 p_{10}(\zeta) + \pi_2 p_{11}(\zeta) \quad (6.3-6)$$

where the $p_{ij}(\zeta) \triangleq p_1[\zeta | \underline{v} = (i,j)]$ are given by (6.1-16)-(6.1-19) and the event probabilities π_0 , π_1 , and π_2 are given by Table 6.2-1.

The density $p_2(\zeta)$ from (6.3-5) contains 10 terms, as discussed in Section 6.1.3.2. Analytical results for three of these cases are given by (6.1-23) and (6.1-24).

By performing the convolution (6.3-3) using (6.3-6) and (6.1-23), we obtain a form containing the sum of seven numerical convolutions. However, each convolution involves a reasonably well-behaved integrand. Overall, the computational effort is also lightened by the reduction in total terms due to splitting up the 3-hop jamming events into 1-hop events in (6.3-6) and 2-hop events in (6.3-5) for which, at least in part, analytical results are available. This is the method implemented by the computer program given in Appendix L.

J. S. LEE ASSOCIATES, INC.

6.4 NUMERICAL RESULTS

The numerical computations for the self-normalizing receiver suffer difficulties similar to those encountered for the clipper receiver, namely a multitude of multiple-dimensional numerical integrations. For the case of $M=2$ and $L=2$, from (6.2-2) and (6.1-16), we have the simplest case to compute, consisting of 7 one-dimensional integrals. For the case of $M=2$ and $L=3$, we must do 5 two-dimensional numerical integrations and 2 three-dimensional numerical integrations to obtain a value of $P_b(e)$ for given E_b/N_0 , E_b/N_J , and γ . For the case of $M=4$ and $L=2$, we are faced with numerical integrations in five or six dimensions over non-standard regions (see, for example, Figure 6.2-1 for the region of integration of the outermost 3 dimensions).

At the computational throughput rate of the PDP-11/44 computer available for the computations, the CPU time to obtain results for even $M=4$ and $L=2$ are estimated to run to many months, or even years. Hence, we restrict our numerical computations to $M=2$ and $L=2$ and 3. For $L=2$, the speed of computation was sufficiently high to permit the full set of curves for $\gamma = 0.001$ to $\gamma = 1.0$, to be computed using the program in Appendix K. Figure 6.4-1 shows the performance as a function of E_b/N_J when $E_b/N_0 = 13.35247$ dB, which corresponds to $P_b(e) = 10^{-5}$ for ideal MFSK with $M=2$. We note in Figure 6.4-1 that there is a clustering of the cross-over around $E_b/N_J = 16$ dB.

If E_b/N_0 is increased to 20 dB, the performance curves shown in Figure 6.4-2 are obtained. In this figure we observe that many of the curves exhibit a breakpoint at which the direction of curvature changes. Clearly, the self-normalizing receiver departs considerably from the ideal receiver under certain ranges of operating conditions. The exact mechanisms which come into play to explain this behavior are not totally clear, but it appears to be the

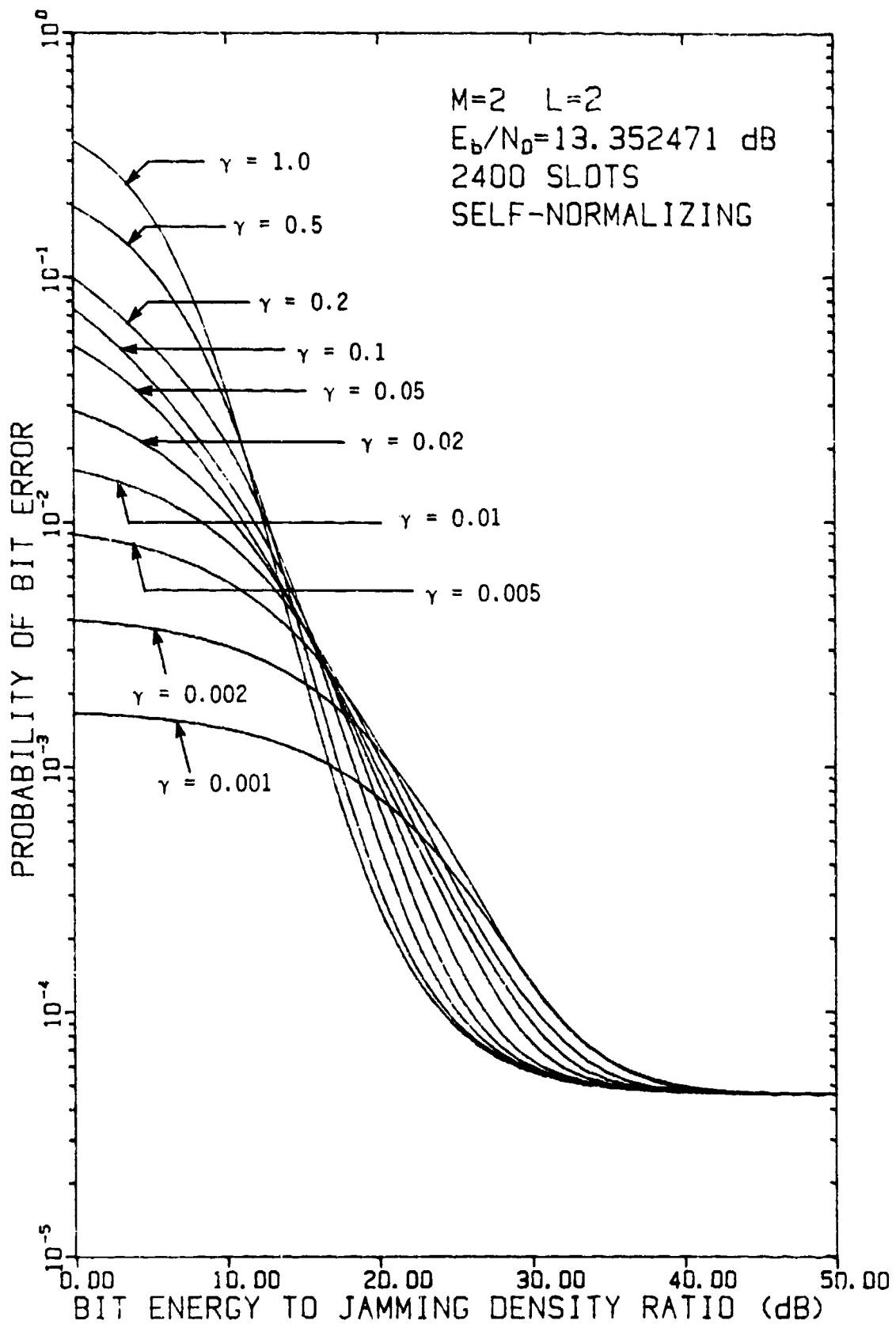


FIGURE 6.4-1 PERFORMANCE OF SELF-NORMALIZING RECEIVER FOR FH/RMFSK WITH $M=2$, $L=2$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.35247$ dB (FOR $P_b(e)=10^{-5}$ IN THE ABSENCE OF JAMMING WHEN $L=1$)

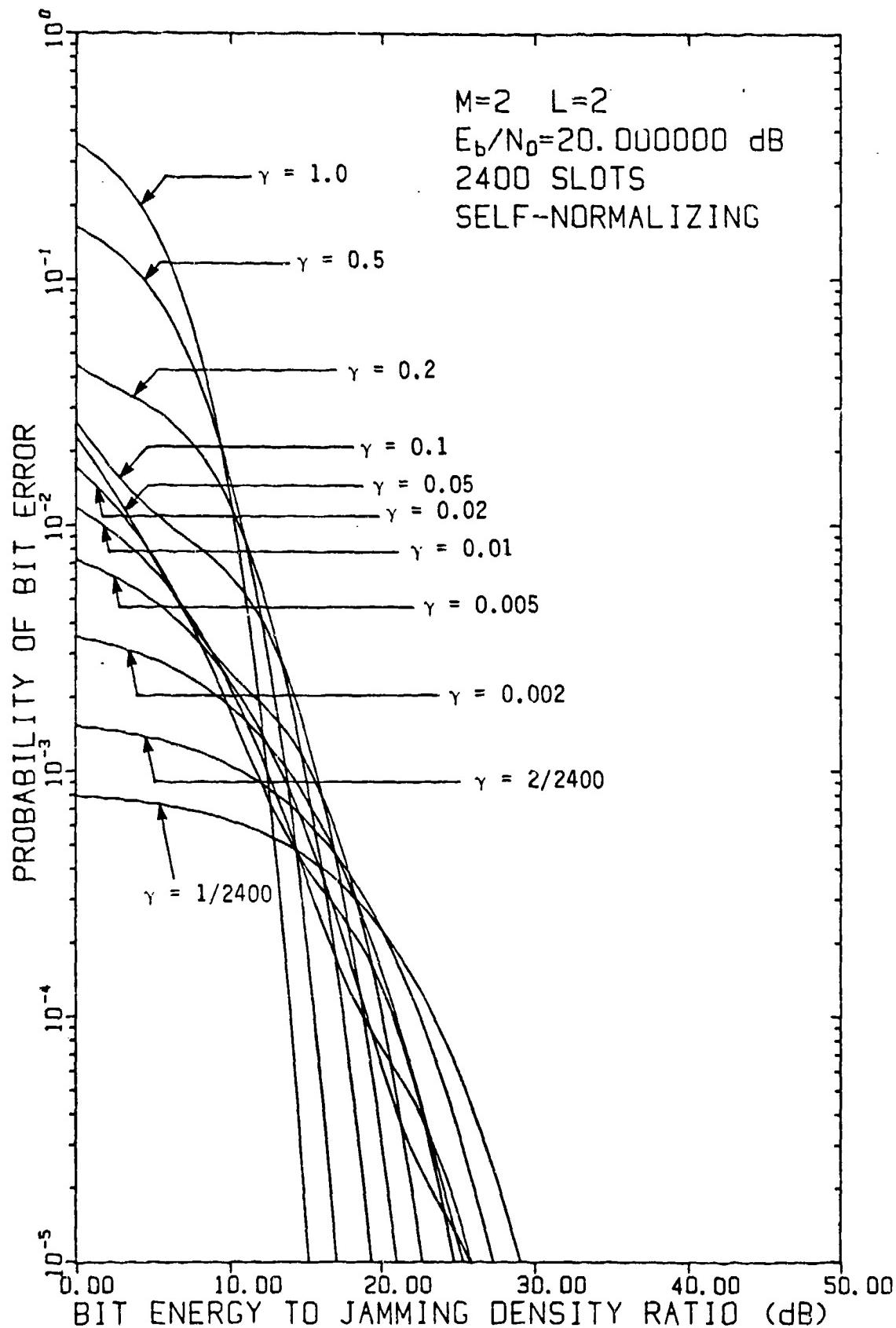


FIGURE 6.4-2 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING NOISE DENSITY RATIO FOR SELF-NORMALIZING RECEIVER IN WORST-CASE PARTIAL-BAND NOISE JAMMING FOR $M=2$, $L=2$ AT $E_b/N_0 = 20$ dB

J. S. LEE ASSOCIATES, INC.

interaction of several different effects, with the switch-over from thermal-noise-limited operation to partial-band-jamming-limited operation playing a significant role. The importance of this switch-over is supported by the lack of apparent breakpoints in the curves for very small γ ($\gamma = 0.001, 0.002$) and very large γ ($\gamma = 0.5, 1.0$); these are the cases in which the one-slot-jammed jamming event predominates and the shape of the curves reflects essentially the performance conditioned on the dominant event.*

For the case of $L=3$ hops, the reduced speed of computation dictated that we search for the optimum jamming fraction at each E_b/N_J rather than run full curves for the various values of γ . Again, we started at $\gamma = 1/2400$ for $E_b/N_J = 50$ dB to speed the search, and then stepped to a lower value of E_b/N_J . The computer program in Appendix L was used to obtain the results presented in Figure 6.4-3 for $M=2, L=3$.

Finally, Figure 6.4-4 compares the performances of the self-normalizing receiver as L , the number of hops per symbol, varies. As we have observed with the other receivers, there is a limited range of E_b/N_J over which a "diversity" effect is achieved. For example, $L=3$ outperforms $L=2$ for $17 \text{ dB} < E_b/N_J < 29 \text{ dB}$. However, in the thermal-noise-limited region and in the strong-jamming region (where $\gamma = 1.0$ is the worst-case jamming), the noncoherent combining loss dominates and $L=1$ is optimum.

* For further discussion, see Section 7.3.3.5.

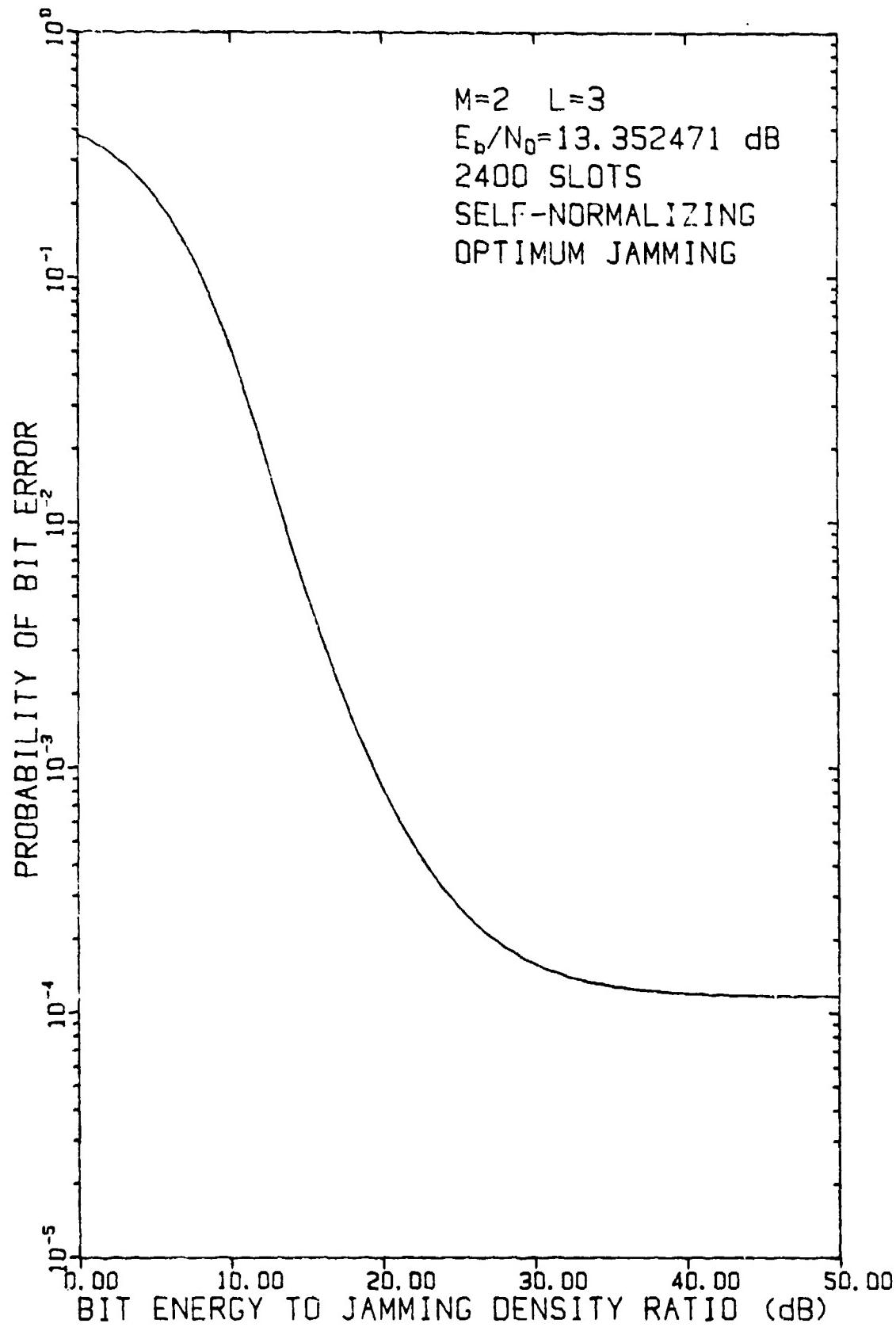


FIGURE 6.4-3 PERFORMANCE OF SELF-NORMALIZING RECEIVER FOR FH/RMFSK WITH $M=2$, $L=3$ HOPS/SYMBOL, 2400 HOPPING SLOTS, AND $E_b/N_0 = 13.35247 \text{ dB}$ (FOR $P_b(e) = 10^{-5}$ IN THE ABSENCE OF JAMMING WHEN $L=1$)

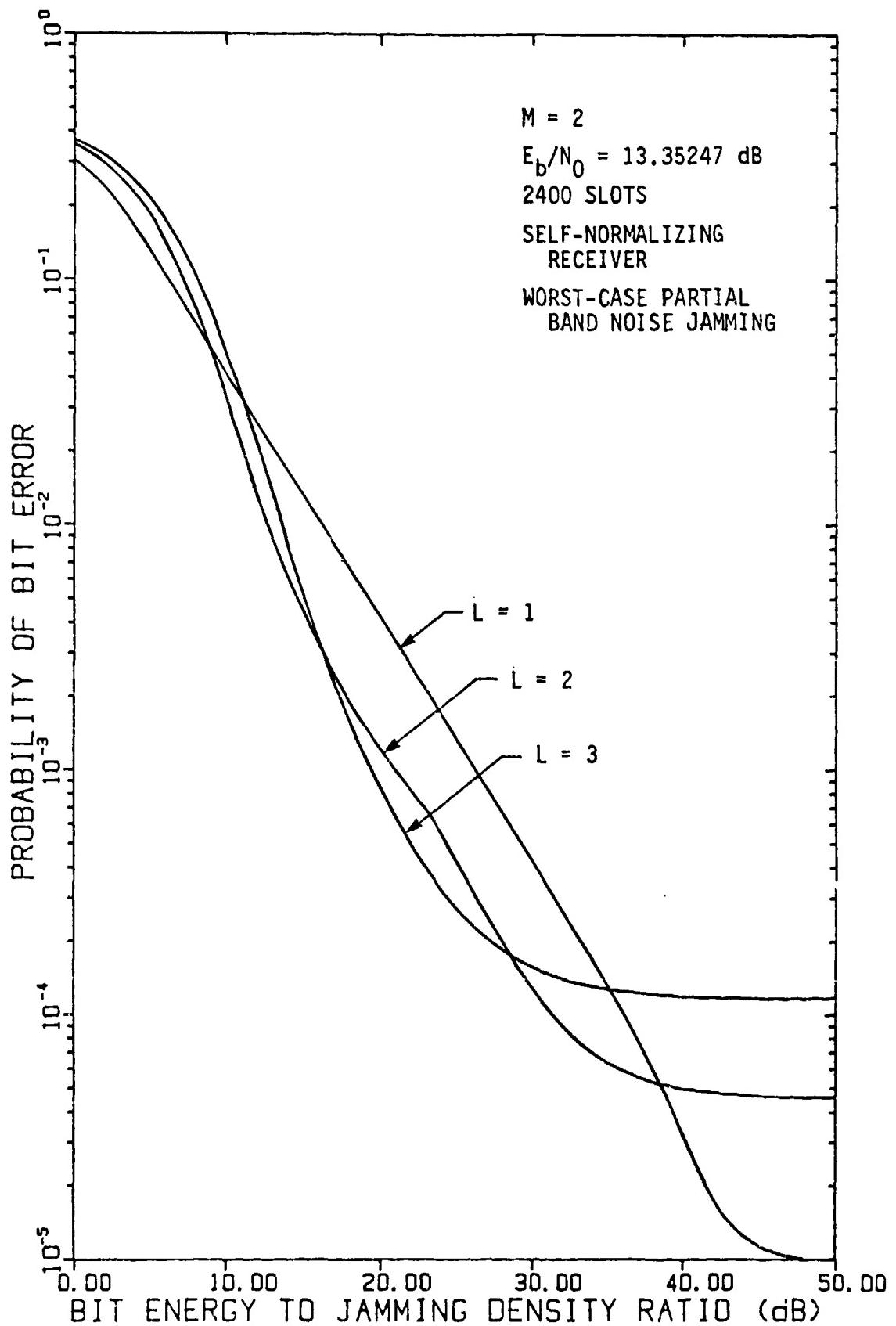


FIGURE 6.4-4 PERFORMANCE OF SELF-NORMALIZING RECEIVER FOR FH/RMFSK WITH $M=2$, AND $E_b/N_0 = 13.35247 \text{ dB}$ WITH NUMBER OF HOPS/SYMBOL AS A PARAMETER

J. S. LEE ASSOCIATES, INC.

7.0

COMPARISONS OF RECEIVER PERFORMANCES

The information we have generated separately on the performances of the various FH/RMFSK receivers in Sections 3-6 can now be compared to learn which ECCM processing scheme is most effective in worst-case partial-band noise jamming. However, since the random MFSK waveform is specifically designed to counter follow-on jamming, we first develop the performances of these receiver processing schemes in follow-on jamming, both for conventional hopping and for random hopping.

7.1 RECEIVER PERFORMANCES IN FOLLOW-ON NOISE JAMMING

7.1.1 Formulation of Follow-on Jamming Analysis: Simple Jammer.

Under follow-on noise jamming (FNJ), it is assumed that on each hop, the jammer places his available power, J watts, in a relatively narrow band centered on the signal's hop frequency. If this band is at least $2(M-1) + 1$ slots wide, then the jammer is guaranteed to jam all M slots of conventional FH/MFSK on every hop*. The hop SNR for FNJ therefore is

$$\rho_T = \frac{S}{\sigma_N^2 + \sigma_J^2} = \frac{E_h}{N_0 + N_J / \gamma_r} = \frac{\log_2 M}{L} \cdot \frac{E_b}{N_0 + N_J / \gamma_r} \quad (7.1-1a)$$

where the jammer spectral density N_J is defined over the entire hopping system bandwidth, $N_J = J/W$, and therefore the effective jamming fraction is

$$\gamma_r = \frac{W_J}{W} \geq \frac{2M-1}{N} . \quad (7.1-1b)$$

In terms of the jamming events indexed by the vector $\underline{\ell} = (\ell_1, \ell_2, \dots, \ell_M)$, where ℓ_m is the number of hops jammed in symbol frequency

*In Section 7.1.2, we consider an "advanced" FNJ which excludes jamming from the signal slot.

J. S. LEE ASSOCIATES, INC.

channel m , for repeat jamming we have

$$\ell_1 = L. \quad (7.1-2)$$

The values of the other $\{\ell_m\}$ depend on the hopping scheme.

7.1.2 Formulation of FNJ Analysis: Advanced Jammer.

The FNJ can avoid helping the communicator by not putting any jammer power in the signal slot. Assuming that this measure is taken, the hop SNR is ρ_N in the signal slot. For maximum effectiveness against FH/MFSK, we also assume that the jammer places half its power in each of the two slots on either side of the intercepted signal, as illustrated in Figure 7.1-1. Thus $\ell_1 = 0$, and for a single hop, for FH/MFSK,

$$\left. \begin{aligned} \Pr\{1 \text{ nonsignal slot jammed}\} &= \frac{2}{M} \\ \Pr\{2 \text{ nonsignal slots jammed}\} &= 1 - \frac{2}{M}. \end{aligned} \right\} \quad (7.1-3a)$$

For random hopping and advanced FNJ,

$$\left. \begin{aligned} \Pr\{0 \text{ nonsignal slots jammed}\} &= \frac{(N-M)(N-M-1)}{(N-1)(N-2)} \\ \Pr\{1 \text{ nonsignal slot jammed}\} &= 2 \frac{(N-M)(M-1)}{(N-1)(N-2)} \\ \Pr\{2 \text{ nonsignal slots jammed}\} &= \frac{(M-1)(M-2)}{(N-1)(N-2)}. \end{aligned} \right\} \quad (7.1-3b)$$

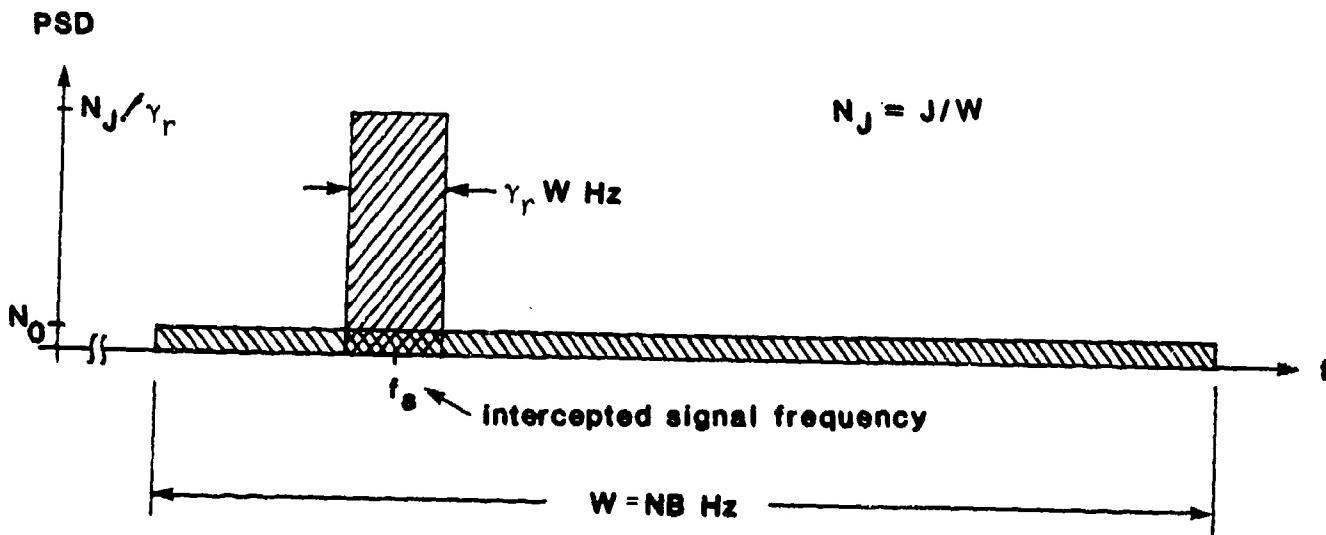
7.1.3 Performance of Conventional FH/MFSK in FNJ.

For conventional FH/MFSK, the M symbol frequency slots are contiguous at RF on each hop. Under simple FNJ, therefore, all M slots are jammed on each hop. That is,

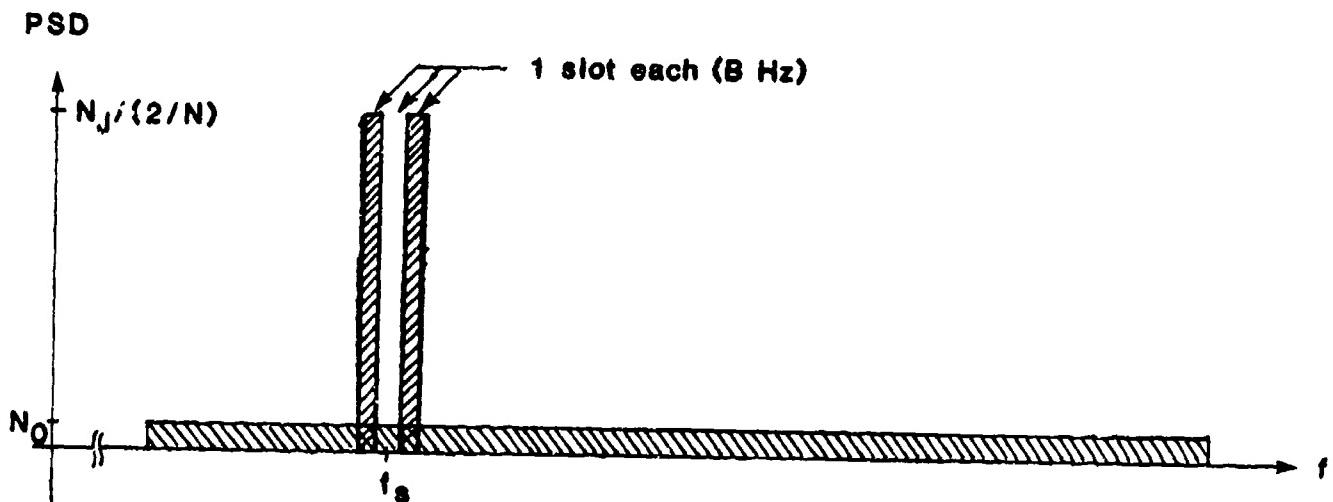
$$\ell_2 = \ell_3 = \dots = \ell_M = L. \quad (7.1-4)$$

The error probability for the system then is

$$P_b(e; \gamma_r) = \frac{M/2}{M-1} P_s(e; \gamma_r | \ell_m = L, \text{ all } m). \quad (7.1-5)$$



(a) "Simple" follow-on jammer.



(b) "Advanced" follow-on jammer.

FIGURE 7.1-1 THERMAL NOISE AND FOLLOW-ON JAMMER MODELS

J. S. LEE ASSOCIATES, INC.

For the several receiver processing schemes we have the results using ρ_T given in (7.1-1),

$$P_b(e; \gamma_r) = \text{probability of error for broadband jamming } (\gamma=1), \\ \text{with } E_b/N_j \text{ replaced by } E_b/N_j + 10 \log_{10} \gamma_r. \quad (7.1-6)$$

Thus the error curves for follow-on jamming, $P_b(e)$ vs. E_b/N_j with $E_b/N_0 = \text{constant}$, are those for full-band jamming, but moved to the right by $-10 \log_{10} \gamma_r$ dB.

Figure 7.1-2 illustrates the performance of conventionally-hopped FH/MFSK in follow-on noise jamming, assuming the follow-on jammer's bandwidth guarantees jamming of all M slots of the symbol. For example, if there are 2400 hopping slots and the jammer occupies 100 slots, then $\gamma_r = 100/2400 = -13.8$ dB, so that the effective E_b/N_j shown in the figure runs from about 14 dB to 64 dB. Something like 25 dB E_b/N_j gives an error rate of 10^{-2} for $L = 1$.

For the advanced FNJ, the effect on FH/MFSK is to produce the symbol error rate

$$P_s = \frac{2}{M} P_s(e | \ell_2=L; \ell_m=0, m \neq 2) \\ + (1 - \frac{2}{M}) P_s(e | \ell_2=\ell_3=L; \ell_m=0, m \neq 2, 3); \quad (7.1-7a)$$

$$\text{with } \gamma_r = 2/N. \quad (7.1-7b)$$

Figure 7.1-3 illustrates the jammed BER for this case for $M=2, 4$, and 8. It is quite clear the the follow-on capability gives the jammer a tremendous advantage against FH/MFSK.

7.1.4 Performance of FH/RMFSK in Follow-on Jamming.

The FH/RMFSK hopping scheme is designed to defeat FNJ by making it difficult for the jammer to jam the nonsignal slots; the nonsignal slots are distributed randomly in the hopping band. The simple follow-on jammer

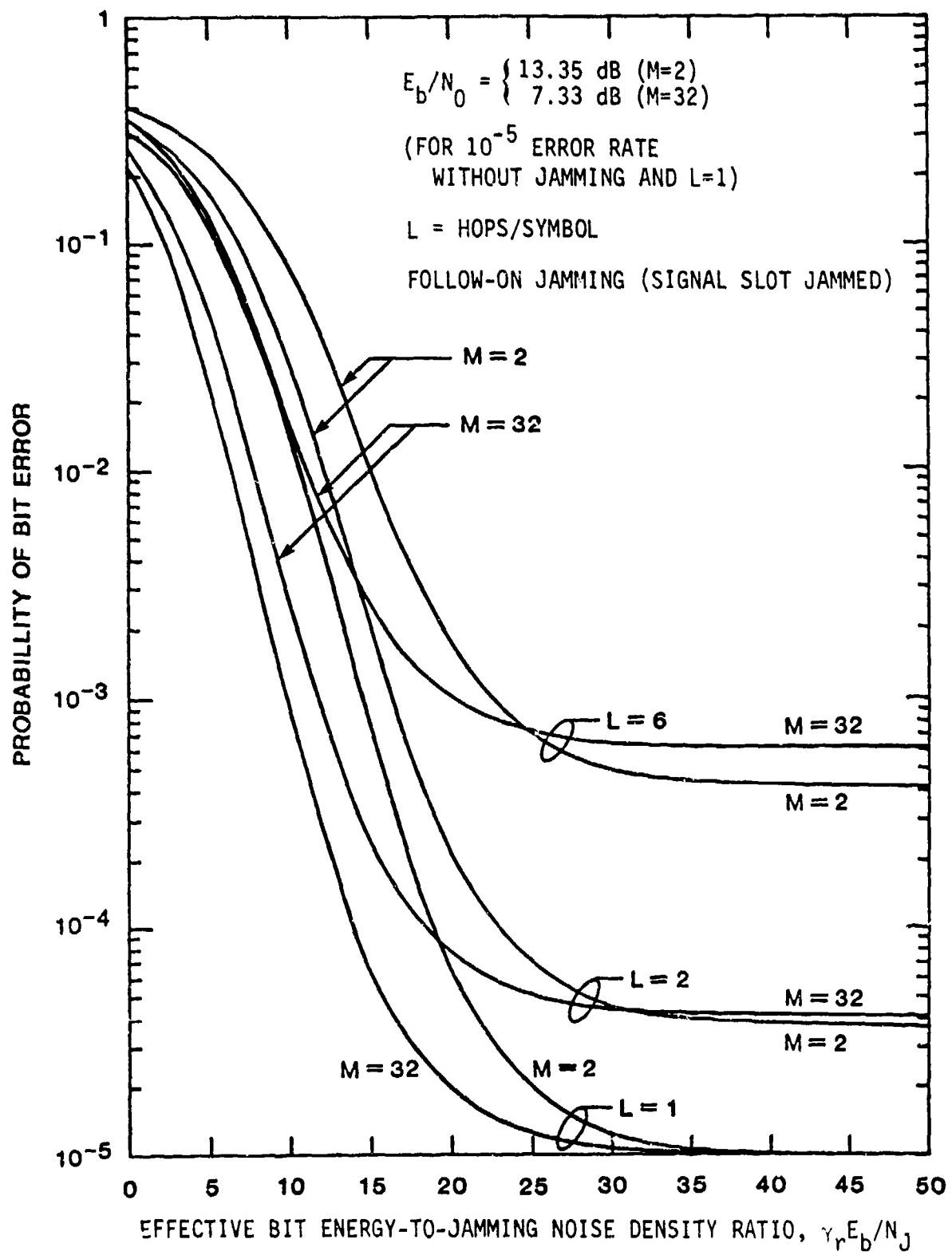


FIGURE 7.1-2 FOLLOW-ON JAMMING PERFORMANCE OF LCR AND ACJ-AGC RECEIVERS FOR FH/MFSK WITH $L=1, 2, 6$ AND $M=2, 32$ WHEN E_b/N_0 YIELDS A 10^{-5} BER FOR $L=1$

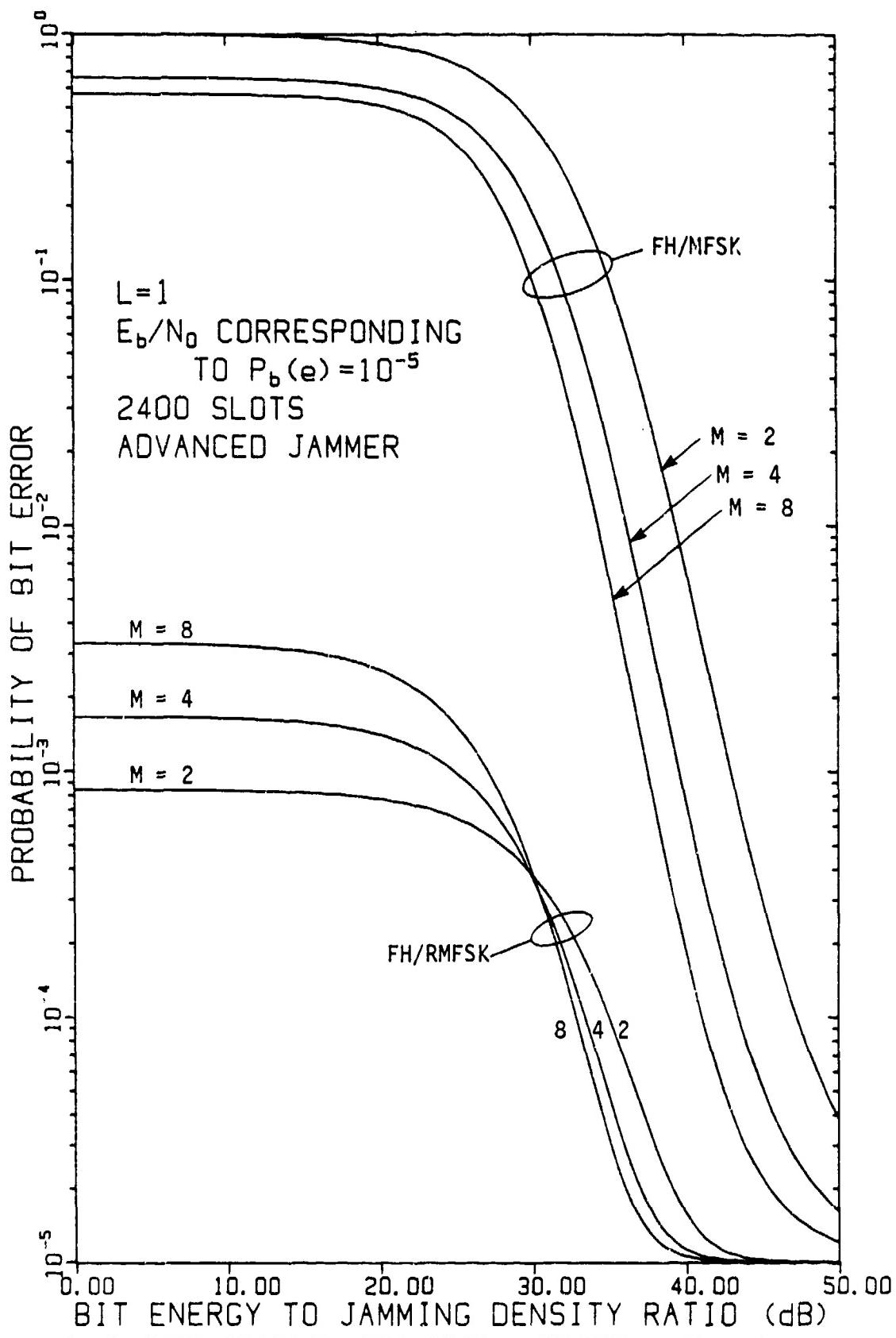


FIGURE 7.1-3 ADVANCED FOLLOW-ON NOISE JAMMING PERFORMANCE OF FH/RMFSK AND FH/MFSK FOR $L = 1$ HOP/SYMBOL AND UNJAMMED BER = 10^{-5}

J. S. LEE ASSOCIATES, INC.

can very likely help the receiver by placing more RF power into the signal's slot.

For $M \ll N$, that is, for a very wide hopping band compared to the symbol bandwidth, the effect of FNJ on FH/RMFSK is approximately to guarantee that

$$l_1 = L$$

$$l_2 = l_3 = \dots = l_M = 0. \quad (7.1-8)$$

This has two effects. First, the per-hop SNR ρ_T is decreased since additional noise is inserted in the signal channel by the jammer. Second, the scaling of the square-law envelope samples increases for the same reason. Since the average value of a single sample is

$$E\{z_{1k}\} = 2(\sigma_T^2 + S),$$

$$E\{z_{mk}\} = 2\sigma_N^2, \quad m \geq 2, \quad (7.1-9)$$

on the whole we anticipate that repeat jamming will increase the probability that the receiver will make a correct decision.

7.1.4.1 Soft-decision receivers.

For the various soft-decision receivers studied, under simple FNJ the decision variables are described as follows:

Linear combining (conventional) receiver

$$z_1 = \sigma_T^2 x^2(2L; 2L\rho_T)$$

$$z_m = \sigma_N^2 x^2(2L), \quad m \geq 2. \quad (7.1-10)$$

AGC - individual channel normalization receiver (IC)

$$z_1 = x^2(2L; 2L\rho_T)$$

$$z_m = x^2(2L), \quad m \geq 2. \quad (7.1-11)$$

J. S. LEE ASSOCIATES, INC.

AGC - any channel jammed normalization receiver (ACJ)

$$z_1 = x^2(2L, 2L\rho_T)$$

$$z_m = \frac{\sigma_N^2}{\sigma_T^2} x^2(2L), \quad m \geq 2. \quad (7.1-12)$$

Clipper receiver

$$\begin{aligned} z_1 &= \sum_{k=1}^L [\sigma_T^2 x^2(2; 2\rho_T)] \text{ clip at } n \\ z_m &= \sum_{k=1}^L [\sigma_N^2 x^2(2)] \text{ clip at } n. \end{aligned} \quad (7.1-13)$$

Since multiplication of all channels by a constant factor does not affect the error probability, we observe that the conventional and AGC-ACJ receivers will achieve the same performance in FNJ:

$$\begin{aligned} P_b(e; \gamma_r)_{ACJ} &= \frac{M/2}{M-1} \sum_{k=1}^{M-1} \binom{M-1}{k} \frac{(1)^{k+1}}{(1+kK)^L} \exp \left\{ -\frac{kKL\rho_T}{1+kK} \right\} \\ &\times \sum_{r=0}^{k(L-1)} C(k, r) \left(\frac{K}{1+kK} \right)^r \mathcal{L}_r^{L-1} \left(\frac{-L\rho_T}{1+kK} \right), \end{aligned} \quad (7.1-14a)$$

where

$$K \triangleq \sigma_T^2 / \sigma_N^2, \quad (7.1-14b)$$

$$C(k, r) = \frac{1}{r} \sum_{n=1}^{\min(r, L-1)} \binom{r}{n} \left[(k+1)n - r \right] C(k, r-n), \quad (7.1-14c)$$

$$C(k, 0) = 1,$$

and $\mathcal{L}_r^{L-1}(\cdot)$ denotes the generalized Laguerre polynomial of degree r with parameter $L-1$.

J. S. LEE ASSOCIATES, INC.

The performance for the AGC receiver with individual channel normalization is the same as given by (7.1-14), but with K=1.

$$P_b(e; \gamma_r)_{IC} = P_b(e; \gamma_r)_{ACJ} \Big|_{K=1}. \quad (7.1-15)$$

The performance of the clipper receiver in follow-on jamming and FH/RMFSK is

$$P_b(e; \gamma_r)_C = 1 - \int_0^{L_n^*} dx f_L(x; L) \left[G_L(x; 0) \right]^{M-1} - \left[Q\left(\sqrt{2\rho_T}, \sqrt{n^*/\sigma_f^2}\right) \right]^L \frac{e^{L_n^*/2\sigma_f^2}}{M} \left[1 - \left(1 - e^{-L_n^*/2\sigma_N^2} \right)^M \right], \quad (7.1-16)$$

where n^* is the optimum clipping threshold. For $L=1$, this threshold is infinite, causing the clipper receiver to have the same performance as the ACJ receiver under follow-on jamming for that case.

Figure 7.1-4 illustrates the performance of the randomly-hopped FH/RMFSK receivers against simple FNJ for $L=1$. We observe that the error probability is maximized for a particular value of $\gamma_r E_b / N_j$; for the binary case, this value is slightly greater than 0 dB, while for $M=4$ and $M=8$, it is approximately -2.5 dB and -4.0 dB, respectively, for the assumed values of E_b / N_0 (chosen to achieve 10^{-5} error rate without jamming). It is interesting to note that for very strong jamming (to the left of the maximum error), the error rate increases with M as does the maximum error. Using the example of the last subsection, a 10^{-2} error rate is achieved for an E_b / N_j of about 17 dB for $\gamma_r \approx -14$ dB, an 8 dB improvement over conventional hopping.

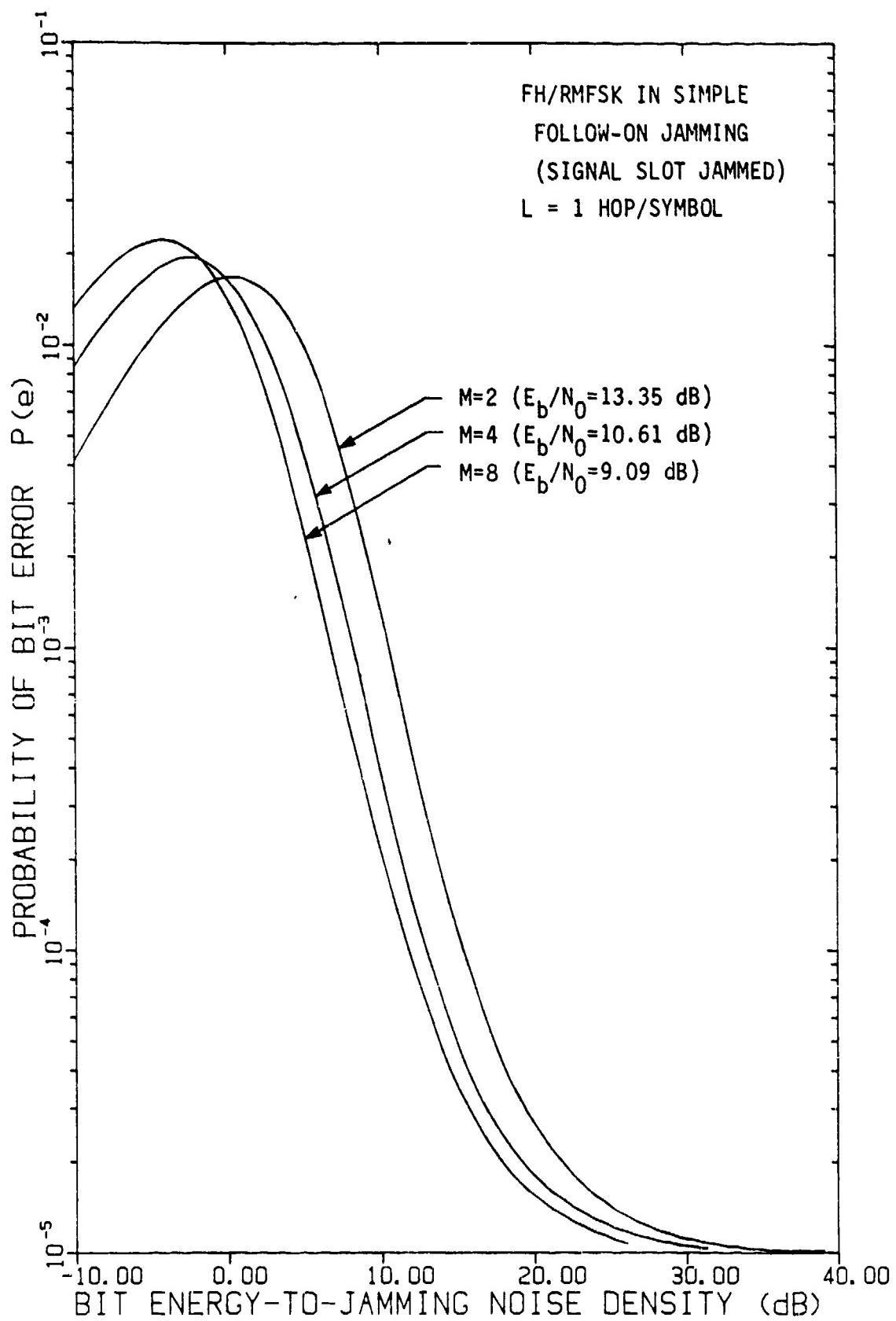


FIGURE 7.1-4 PERFORMANCE OF FH/RMFSK RECEIVERS IN SIMPLE FOLLOW-ON NOISE JAMMING VS. EFFECTIVE BIT ENERGY TO JAMMING NOISE DENSITY RATIO FOR $L=1$ HOP/SYMBOL AND $M=2,4,8$ WHEN E_b/N_0 YIELDS A 10^{-5} BER WITHOUT JAMMING

J. S. LEE ASSOCIATES, INC.

Figure 7.1-5 demonstrates that the same critical jammer power effect is observed for $L=2$, but at different values (approximately 3 dB higher), so that the jammer must know L in order to be effective. It is also evident from this figure that the maximum error rate is decreased by increasing L to 2.

For the advanced FNJ described above, using RMFSK hopping, there is only a slight chance of jamming the symbol on a given hop. The possible jamming events for a single hop are the vectors

$$\underline{v}_k = (0, v_{2k}, v_{3k}, \dots, v_{Mk}) \quad (7.1-17)$$

with probabilities

$$Pr\{r \text{ nonsignal slots jammed}\}$$

$$= \tau_r = \frac{\binom{N-M}{2-r}}{\binom{N-1}{2}} \binom{M-1}{r}. \quad (7.1-18)$$

For $M=2$ the result is

$$P_b(e) = \sum_{k=0}^L \binom{L}{k} \left(\frac{2}{N-1}\right)^k \left(1 - \frac{2}{N-1}\right)^{L-k} P_b(e | \ell_1=0, \ell_2=k). \quad (7.1-19a)$$

$$= P_b(e | \ell_1=\ell_2=0) + \frac{2L}{N-1} P_b(e | \ell_1=0, \ell_2=1), \quad N-1 \gg 2. \quad (7.1-19b)$$

Thus the FH/RMFSK hopping scheme achieves very nearly the unjammed error performance of MFSK when the follow-on noise jammer is configured against FH/MFSK. As (7.1-19b) shows, for $L=1$ and $M=2$ the unjammed error rate is increased by, at most, $2/(N-1)$; this quantity equals 8.3×10^{-4} for $N=2400$, and 7.8×10^{-3} for $N=256$. For $L=1$, Figure 7.1-3 shows the performance of FH/RMFSK much improved over FH/MFSK in advanced FNJ.

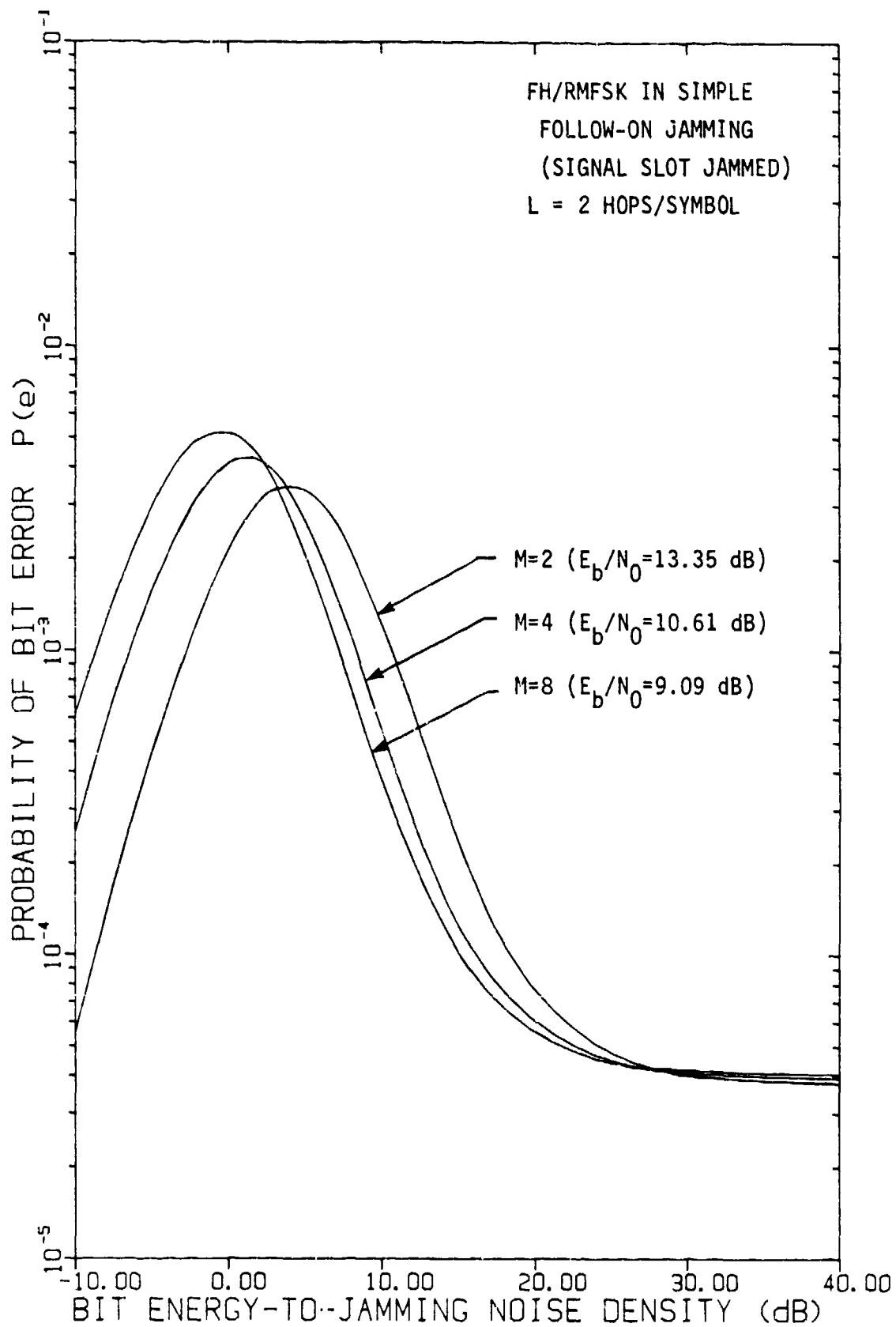


FIGURE 7.1-5 PERFORMANCE OF FH/RMFSK RECEIVERS IN SIMPLE FOLLOW-ON NOISE JAMMING VS. EFFECTIVE BIT ENERGY TO JAMMING NOISE DENSITY RATIO FOR $L=2$ HOPS/SYMBOL AND $M=2, 4, 8$ WHEN E_b/N_0 YIELDS A 10^{-5} BER WITHOUT JAMMING

J. S. LEE ASSOCIATES, INC.

An exception to Figure 7.1-3 for FH/RMFSK is the IC-AGC receiver, for which the "advanced" FNJ is completely nullified. The jammer in this case is "too smart," because the IC-AGC receiver is vulnerable to jamming only if the signal channel is jammed. This statement also holds for the case of conventional MFSK hopping if individual-channel normalization is employed.

7.1.4.2 Hard-decision receiver.

Performance of the hard-decision receiver in simple follow-on noise jamming is depicted in Figures 7.1-6 through 7.1-8 for values of $M=2$, 4 , and 8 respectively. The parameter E_b/N_0 was chosen to yield a 10^{-5} BER in the absence of jamming per respective M value for $L=1$ hop per symbol. It is clearly seen in each of the L $P(e)$ curves that as the jammer power is increased below that E_b/N_j value to cause maximum $P(e)$, a decrease in $P(e)$ takes place. Hence, for strong jamming the jammer is actually aiding the communicator by the addition of energy to the non-coherent FSK signal slot. We also have a diversity improvement for $L \geq 3$ hops per symbol in the strong jamming regions. Conversely, for weak jamming (beyond $E_b/N_j \approx 20$ dB) no diversity improvement is realized for $L \geq 2$ hops per symbol due to the dominance of the noncoherent combining loss existing for the stated thermal noise (E_b/N_0) values. Therefore, in order to be effective (maximum $P(e)$), the jammer must maintain E_b/N_j to within small ranges. For example, to ensure a minimum $P(e)$ of 10^{-2} for $M=2$ and $L=1$ (Figure 7.1-4), E_b/N_j must be held to values ranging from 4 to 14 dB.

The effects of decreased thermal noise levels ($E_b/N_0 = 20$ dB) for cases of $M=2$, 4 , and 8 are illustrated in Figures 7.1-9 through 7.1-11 respectively. Here we observe all of the L $P(e)$ curves exhibiting a

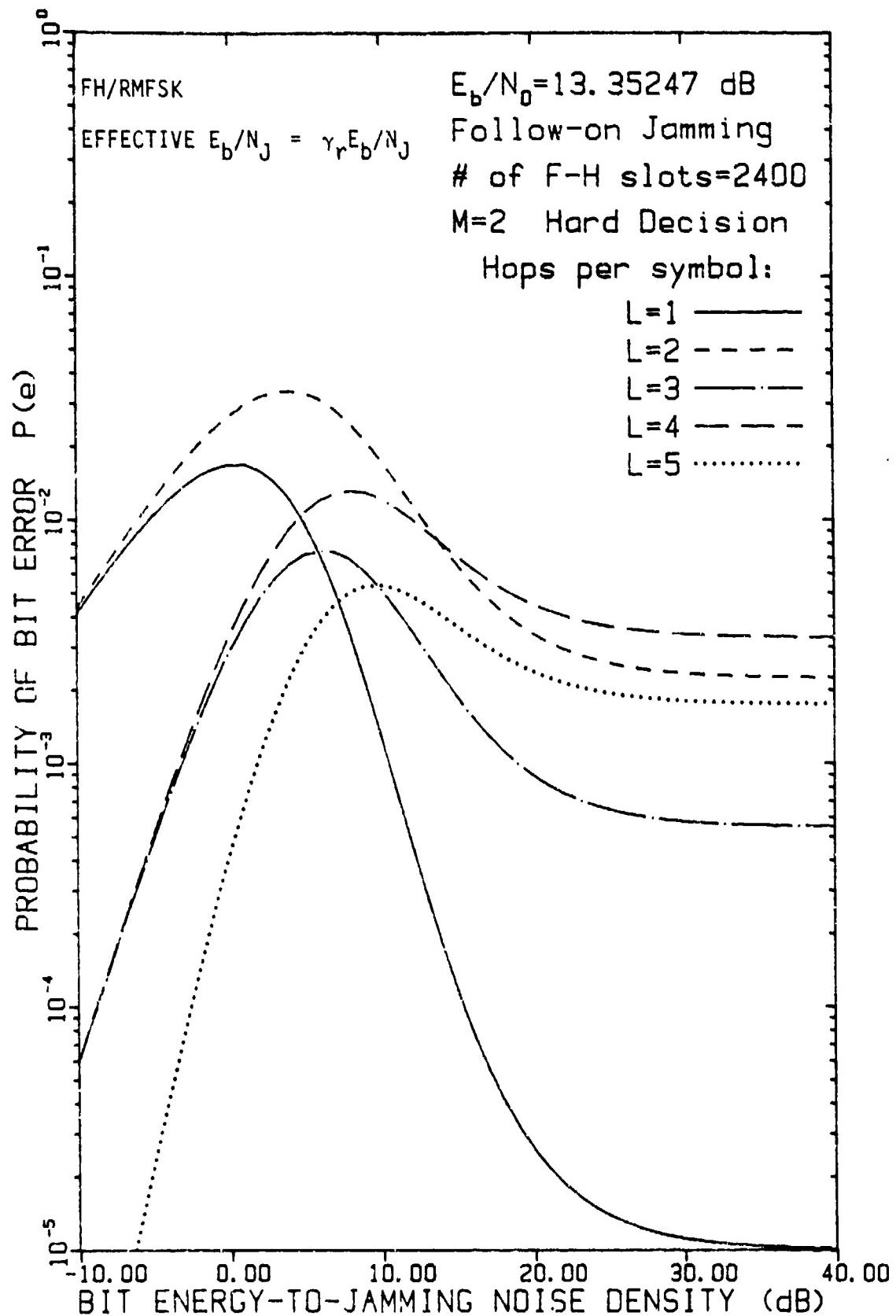


FIGURE 7.1-6 PERFORMANCE OF HARD-DECISION FH/RMFSK RECEIVER IN SIMPLE
 FNJ VS. EFFECTIVE E_b/N_J FOR M=2 AND E_b/N_0 SUCH THAT UNJAMMED
 BER IS 10^{-5} WHEN L=1

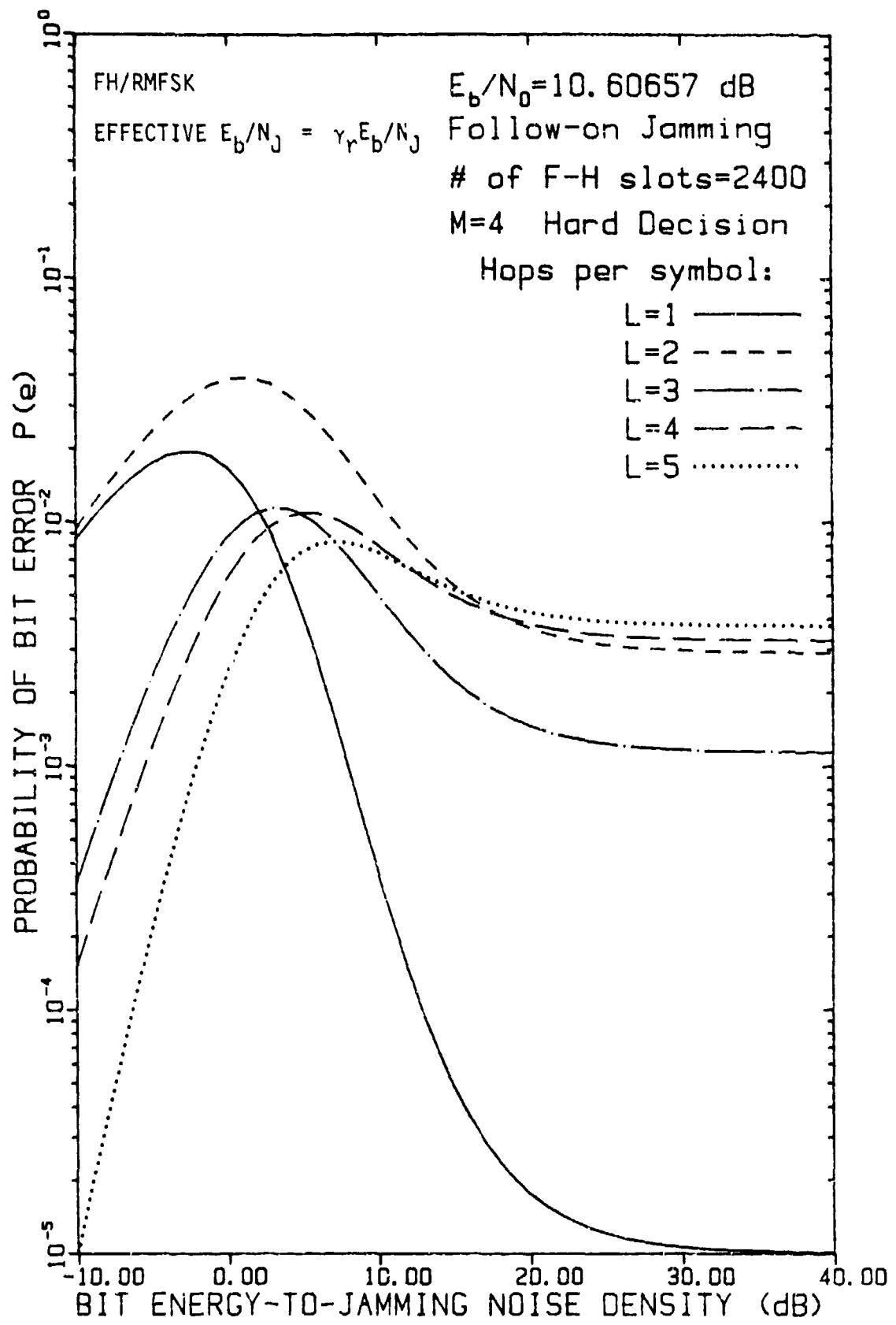


FIGURE 7.1-7 PERFORMANCE OF HARD-DECISION FH/RMFSK RECEIVER IN SIMPLE FNJ VS. EFFECTIVE E_b/N_j FOR $M=4$ AND E_b/N_0 SUCH THAT UNJAMMED BER IS 10^{-5} WHEN $L=1$

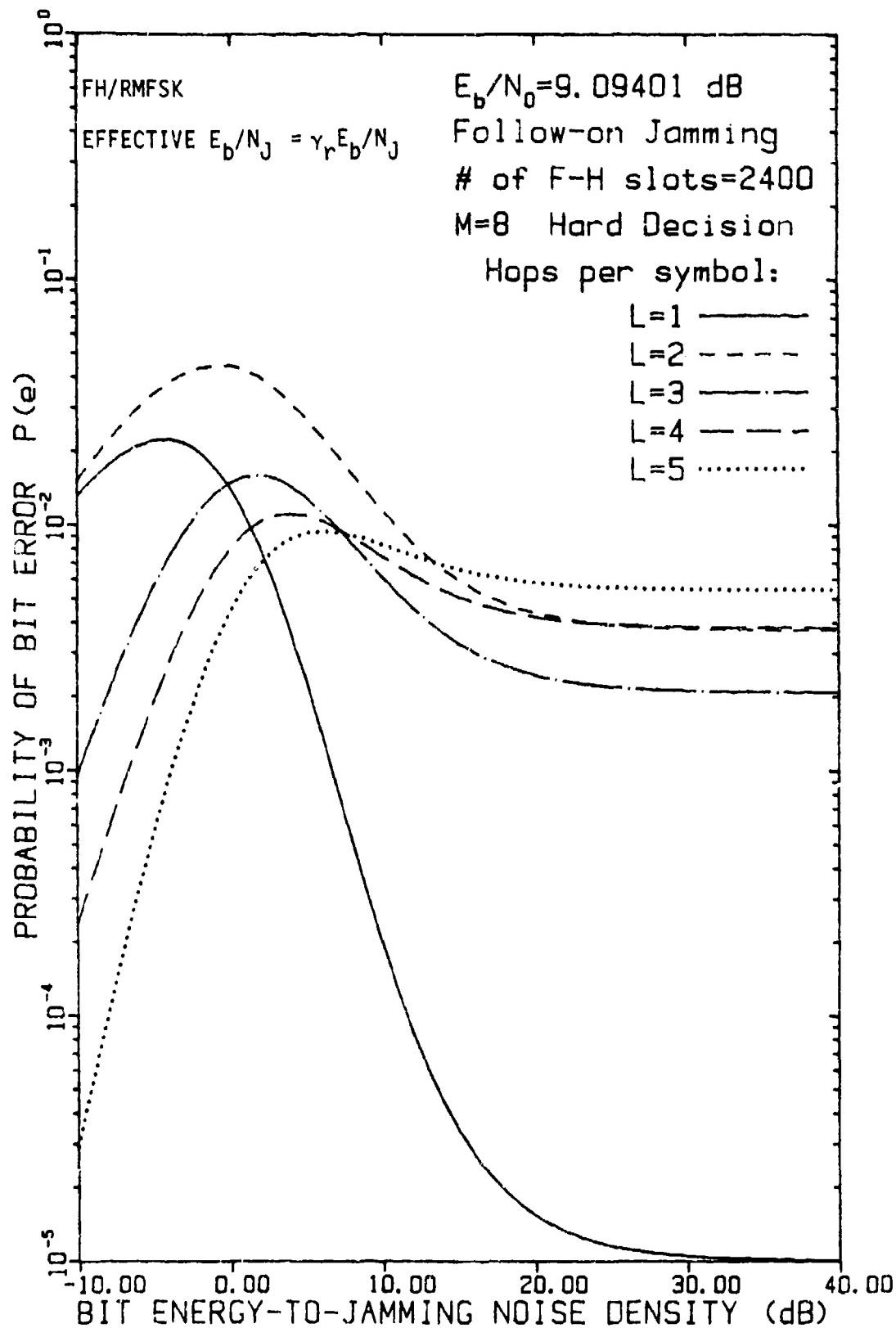


FIGURE 7.1-8 PERFORMANCE OF HARD-DECISION FH/RMFSK RECEIVER IN SIMPLE
 FNJ VS. EFFECTIVE E_b/N_J FOR M=8 AND E_b/N_0 SUCH THAT UNJAMMED
 BER IS 10^{-5} WHEN L=1

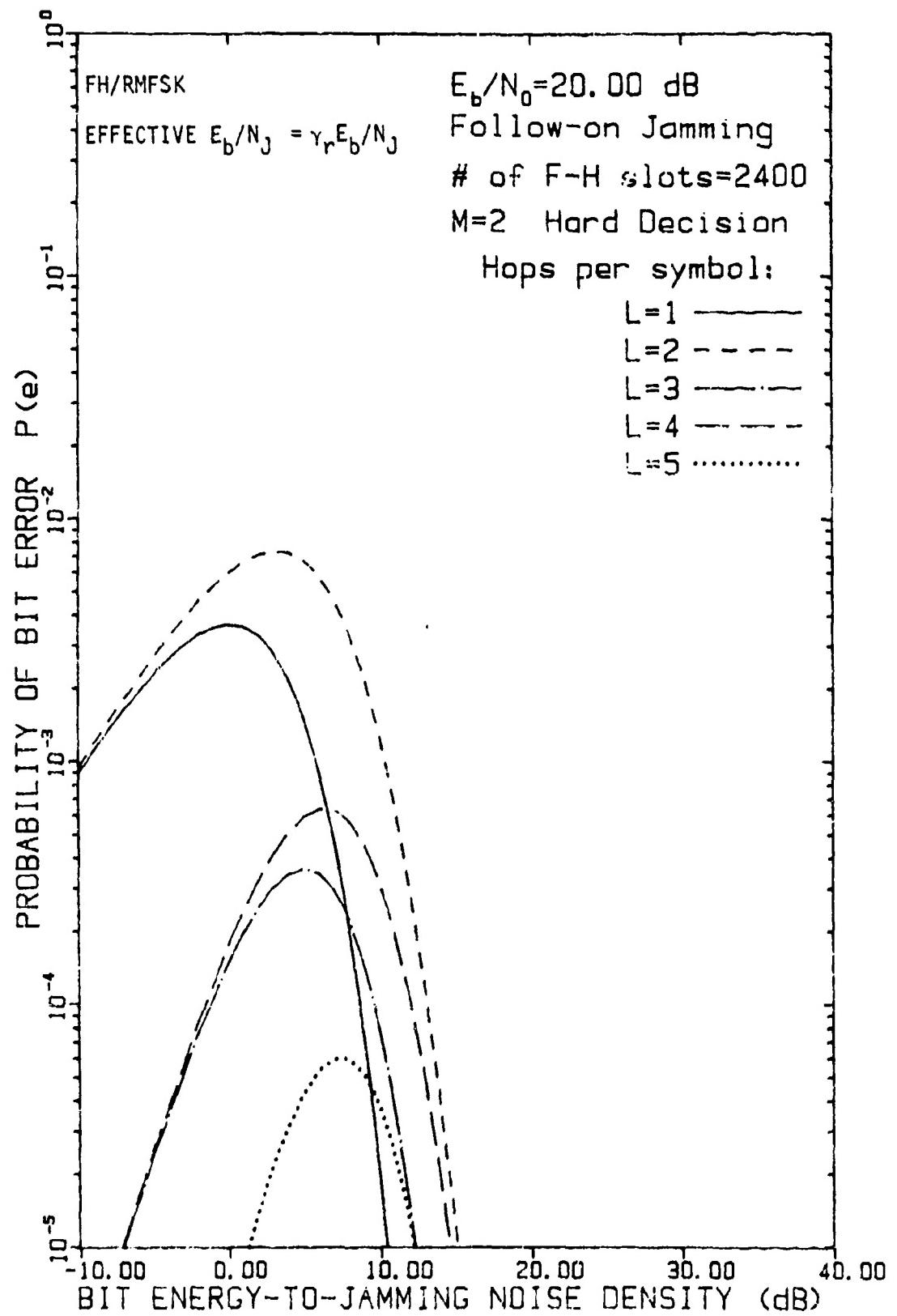


FIGURE 7.1-9 PERFORMANCE OF HARD-DECISION FH/RMFSK RECEIVER IN SIMPLE FNJ VS. EFFECTIVE E_b/N_J FOR M = 2 AND $E_b/N_0 = 20$ dB

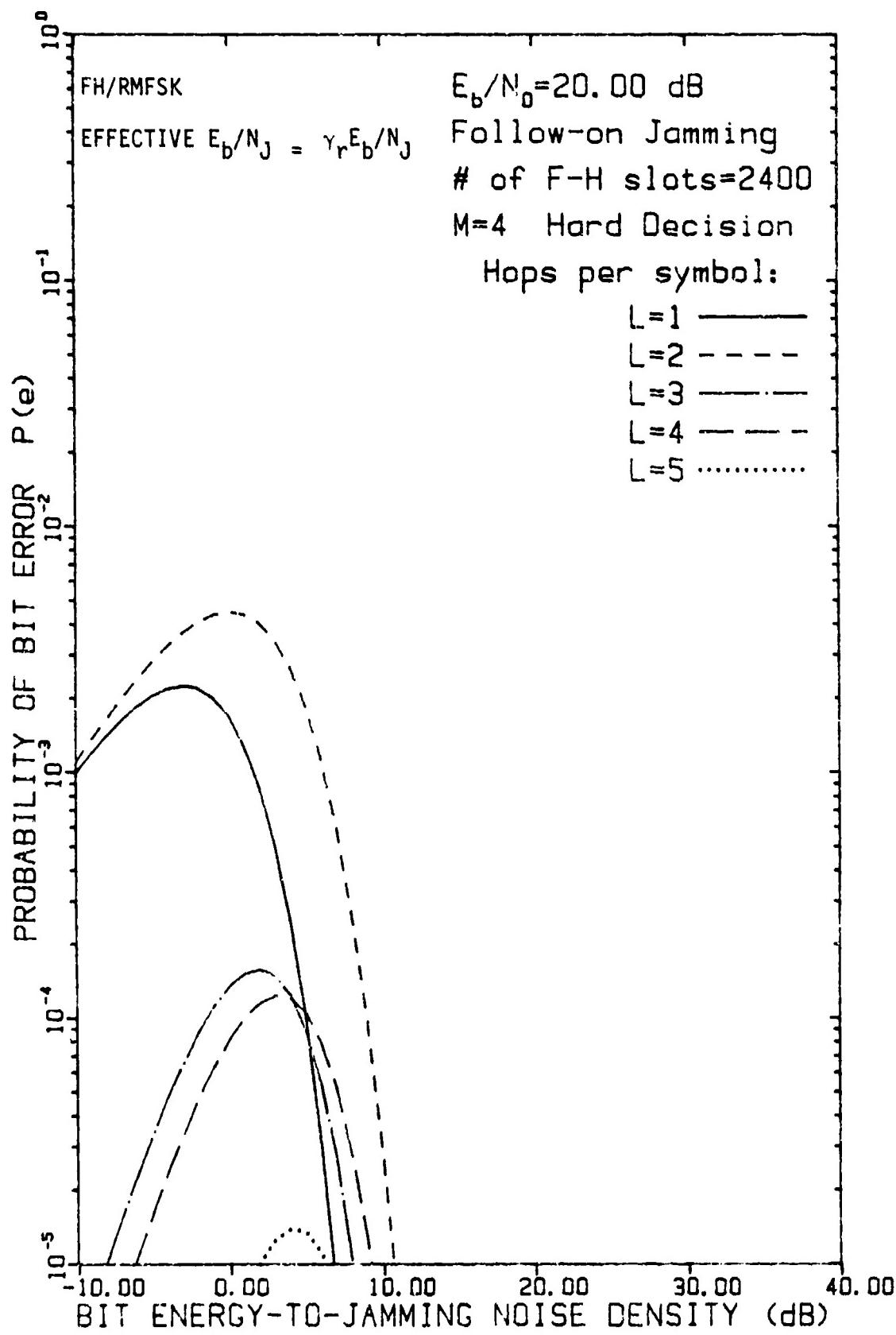


FIGURE 7.1-10 PERFORMANCE OF HARD-DECISION FH/RMFSK RECEIVER IN SIMPLE FNJ VS. EFFECTIVE E_b/N_J FOR M=4 AND $E_b/N_0 = 20$ dB

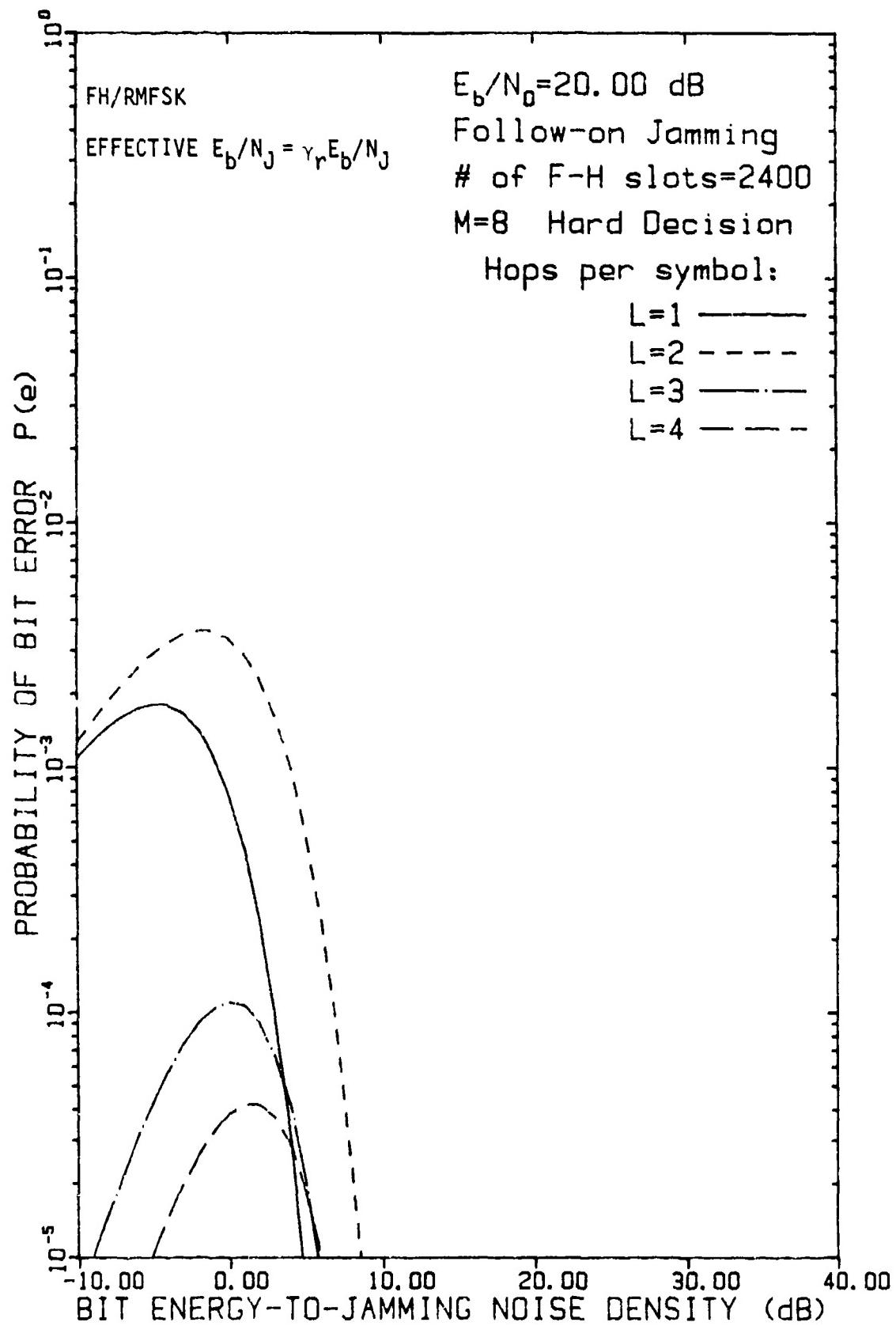


FIGURE 7.1-11 PERFORMANCE OF HARD-DECISION FH/RMFSK RECEIVER IN SIMPLE F NJ
VS. EFFECTIVE E_b/N_J WHEN M=8 AND $E_b/N_0 = 20$ dB

J. S. LEE ASSOCIATES, INC.

"parabolic" type of behavior, i.e. steeply defined strong and weak jamming regions. Thus, the jammer appears to have quite a narrow window of E_b/N_J values to work within for attaining a maximum effect.

7.2 COMPARISONS OF FH/RMFSK RECEIVER PERFORMANCES IN WORST-CASE PARTIAL-BAND NOISE JAMMING (WCPBNJ)

It is understood that the motivation for using the proposed FH/RMFSK waveform is that it is less vulnerable to follow-on noise jamming (FNJ) or repeat noise jamming than is a conventional FH/MFSK block-hopping system. RMFSK is effective in that the FNJ is not able to place jamming power in the unused symbol frequency slots as is the case for MFSK where the M signalling frequencies are adjacent. At the FH/MFSK or FH/RMFSK receiver, the L hops comprising the MFSK symbol can be combined in a number of ways. Certain types of nonlinear combining soft-decision schemes, which weight the detected hops in some form to discriminate against jammed hops, are employed. Previous results [1] have shown that conventional FH/MFSK system performance with L-hop diversity in WCPBNJ is improved by nonlinear combining techniques. This study has addressed FH/RMFSK system performance in the less sophisticated, yet more pervasive, ECM tactic of PBNJ - a basic jamming threat which is inevitably encountered in an EW scenario. In Sections 3 through 6 it has been demonstrated that nonlinear combining yields improved RMFSK performance in this type of jamming.

In what follows, we compare performances of the different types of ECCM receivers for FH/RMFSK signals in the WCPBNJ environment. In Section 7.3 we also compare FH/MFSK and FH/RMFSK receiver performances in WCPBNJ, and in Section 7.4 consider the different effects of the RMFSK diversity combining

J. S. LEE ASSOCIATES, INC.

techniques studied. Unless stated otherwise, we note that all E_b/N_0 values utilized in performance plots are chosen so as to yield a 10^{-5} BER in the absence of jamming when $L=1$ hop per symbol.

Comparisons among the different FH/RMFSK receivers analyzed are provided by Figures 7.2-1 and 7.2-2 for $M=2$ and $E_b/N_0 = 13.35$ dB, and by Figures 7.2-3 and 7.2-4 for $M=4$ at $E_b/N_0 = 10.61$ dB. The enormous amount of computer time required to obtain performance results for the clipper receiver at $L=3$ and the SNORM receiver for $M=4$ is beyond the scope of this study. Therefore, in some figures for comparison, these receivers are not represented. Explanations of the difficulties involved in such calculations were presented in the numerical results of Section 5 (clipper receiver) and Section 6 (SNORM receiver).

We can develop a performance ranking for these receivers with their respective parameter sets (M,L values) by assessing performances in the regions of strong, moderate, and weak jamming. These arbitrary regions are taken to mean the following: (1) strong jamming - usually full-band jamming with E_b/N_j values less than about 4 to 8 dB, (2) weak jamming - region of very small γ values with E_b/N_j usually greater than 35 to 40 dB, and (3) moderate jamming - area between strong and weak jamming.

For the case of $M=2$, $L=2$ (Figure 7.2-1), we find receiver performances asymptotically approaching two groups in the strong jamming region. These are (1) lower $P(e)$: IC-AGC, ACJ-AGC, clipper, and SNORM; (2) higher $P(e)$: hard-decision (HD), and square-law linear combining receiver (LCR). We observe that the first group is more effective due to their nonlinear weighting (normalization) schemes. In the second group, we have the LCR

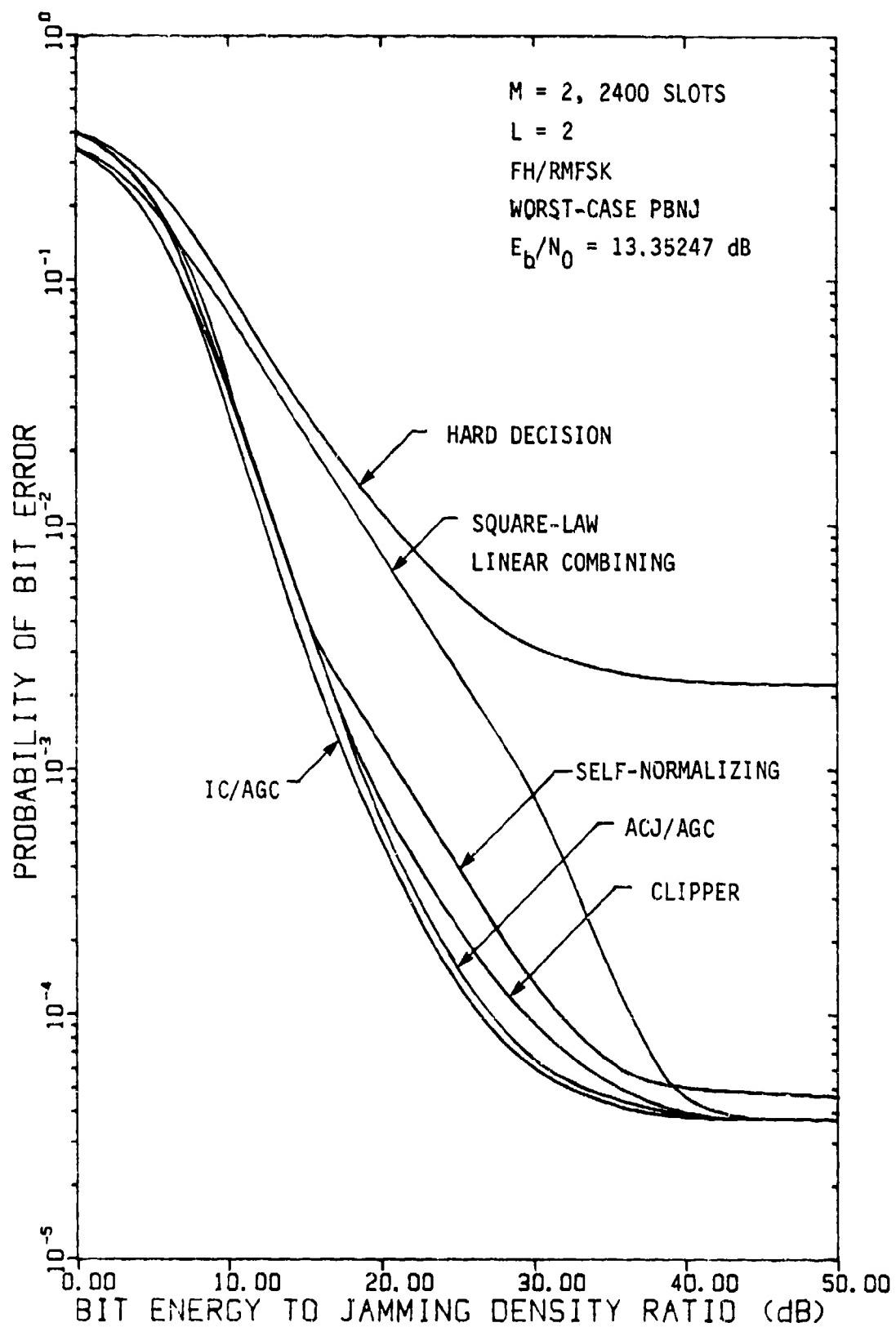


FIGURE 7.2-1 COMPARISON OF DIFFERENT RECEIVERS FOR FH/RMFSK WITH $M=2$ AND $L=2$
HOPS/SYMBOL WHEN $E_b/N_0 = 13.35247$ dB

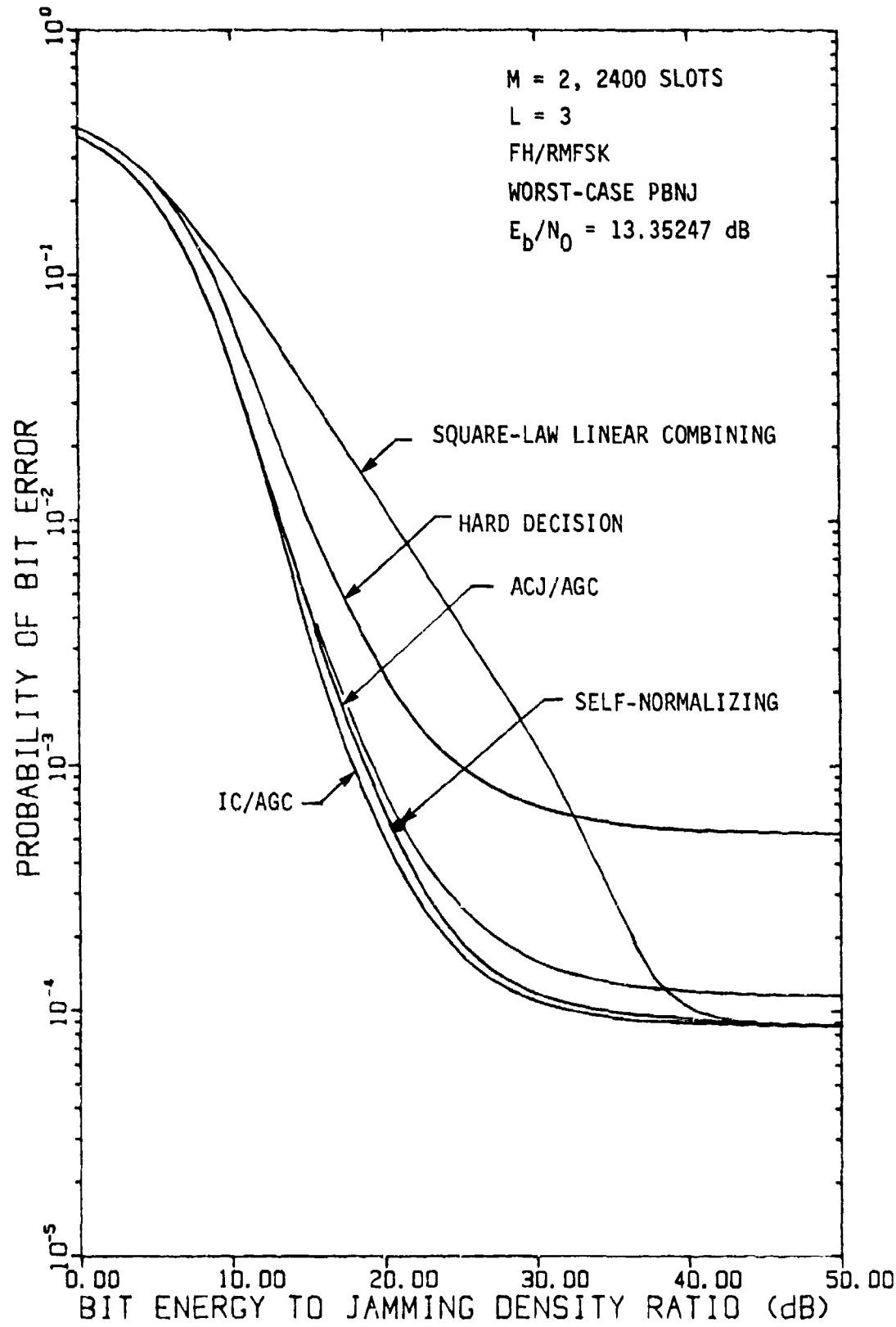


FIGURE 7.2-2 COMPARISON OF DIFFERENT RECEIVERS FOR FH/RMFSK WITH M=2 AND L=3
 HOPS/SYMBOL WHEN $E_b/N_0 = 13.35247$ dB

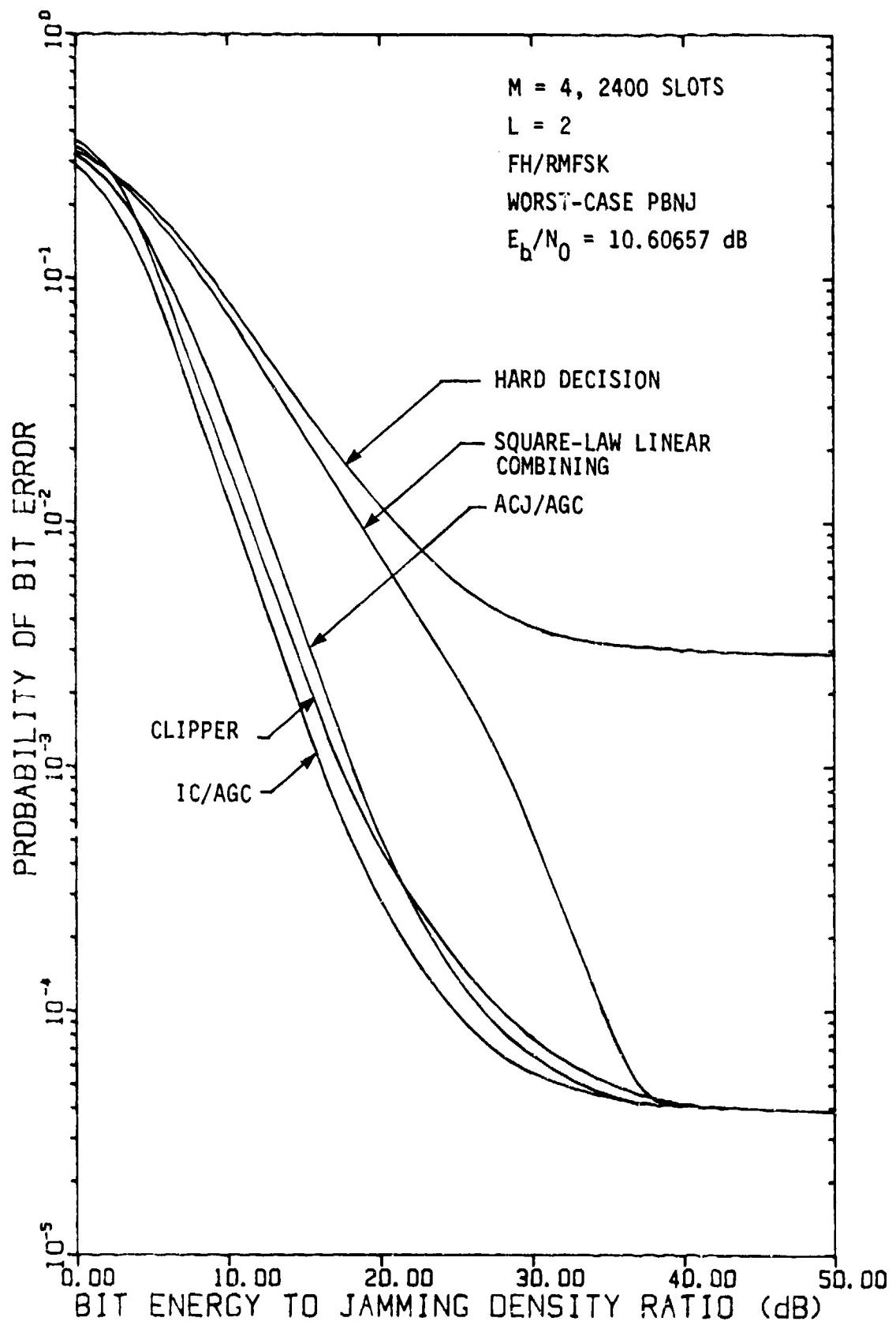


FIGURE 7.2-3 COMPARISON OF DIFFERENT RECEIVERS FOR FH/RMFSK WITH $M=4$ AND $L=2$ HOPS/SYMBOL WHEN $E_b/N_0 = 10.60657$ dB

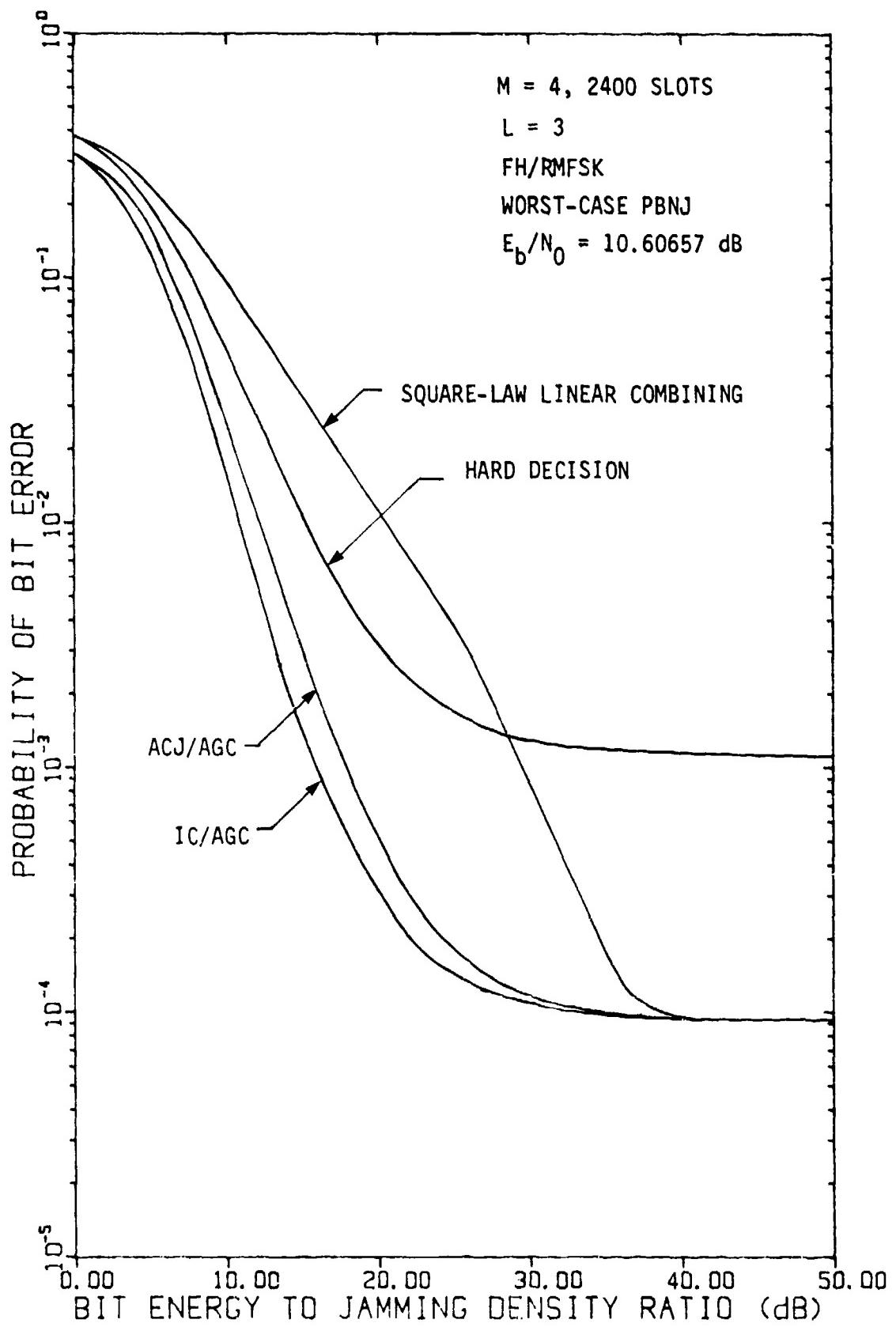


FIGURE 7.2-4 COMPARISON OF DIFFERENT RECEIVERS FOR FH/RMFSK WITH M=4 AND L=3 HOPS/SYMBOL WHEN $E_b/N_0 = 10.60657$ dB

J. S. LEE ASSOCIATES, INC.

which provides no weighting and the HD receiver which is ineffective for $L=2$ because it is subject to the possibility of a tie (for L even) in the final quantized decision variable values. In strong jamming, all receivers are experiencing full-band jamming at high power levels and discriminating against a jammed hop is not done by any receiver since all hops are jammed. It is only when the optimum γ values begin to fall below full-band jamming that we realize the performance improvement of the nonlinear combining techniques. In the moderate jamming region, we see a sub-division among the receivers which we termed "effective" for strong jamming: (a) "ideal" receivers (AGC) and (b) "practical" receivers (clipper, SNORM). In this region we notice performance results for the nonlinear combining types remaining within about 1 dB of each other up until around the point where $E_b/N_J > E_b/N_0$. That is, where thermal noise becomes more dominant than jamming noise. At these values, the SNORM and clipper receiver performances begin to degrade relative to the AGC types, yet still remain superior to the square-law LCR. We note the AGC receivers as maintaining a continuous and graceful transition in this moderate jamming region with the IC-AGC showing a slightly better performance. The worse performance for the ACJ-AGC is due to an imbalancing effect of the ACJ normalization scheme in which all receiver channels are inversely weighted by the largest of all the M channel noise powers. In contrast, the IC-AGC receiver normalizes each channel separately and thereby "balances" or "equalizes" the received noise powers.

The performance "breakaway" for $N_0 > N_J$ for the clipper and SNORM receivers reflects the dominance of the "unbalanced symbol" error mechanism

J. S. LEE ASSOCIATES, INC.

as γ becomes smaller; jamming in more than one M-ary channel becomes unlikely. In the case of the clipper, the receiver tends to limit the unbalancing contribution to the sum without affecting the signal channel sum. But for the SNORM receiver, any input noise power unbalance due to one channel being jammed reduces the signal channel sum. This is because the SNORM normalization weight is inversely proportional to the total noise power (i.e. sum of all M channels) measurement on a given hop without recognition of which individual channels are jammed.

In the weak or no jamming region, all receivers suffer degradation due to the noncoherent combining loss (NCL) when $L > 1$. As E_b/N_j approaches 50 dB (practically no jamming), the different performances of the receivers in the Gaussian channel are evident. All receivers suffer degradation relative to the $L=1$ result ($P(e) = 10^{-5}$) due to the NCL as Figure 7.2-1 demonstrates. We see that the SNORM and HD receivers are subject to higher NCL due, respectively, to inefficient combining and to the possibility of "tie votes" for $M=2$ and $L=2$, with the HD receiver being more severely affected because of its use of only two levels of quantization in the soft-decision.

Figure 7.2-2 compares receiver performances for the parameter set $M=2$, $L=3$. For strong jamming, we see a change in two groupings recognized for the case $M=2$, $L=2$. The HD receiver is now more effective than the square-law LCR, and provides a significant improvement in strong jamming. We attribute this to the fact that no ties exist in the majority logic decoding when L is odd. But in the weak or no jamming region the HD performance is worse than

J. S. LEE ASSOCIATES, INC.

LCR due to quantization noise effects. We likewise note that in moderate jamming, the SNORM performance has improved somewhat over the results for M=2, L=2. It is reasoned that for L=3, the multiple unbalancing effects predominant for decreasing γ values begin to become less probable. Performance rankings in the thermal-noise-limited region for SNORM remain unchanged from the case of M=2, L=2.

The effect of an increase in alphabet size (M=4) can be discerned from Figures 7.2-3 and 7.2-4. We notice that the parameter set M=4, L=2 (Figure 7.2-3) appears similar to the M=2, L=2 set where two distinct groupings are present in moderate jamming with the nonlinear combining soft-decision receiver group yielding superior performance. Considering the moderate jamming region to be from $E_b/N_J = 5$ to 39 dB, we observe that the clipper and ACJ-AGC receivers trade rankings around the regional midpoint of 22 dB; the clipper receiver showing ≤ 1 dB better performance over the range of 5 to 17 dB. However, in strong jamming we find the clipper's performance degrading to the point of being the overall worst performer at $E_b/N_J = 0$ dB. For the HD receiver, we find a worse performance than for M=2, L=2 because there are now two more channels allowing for the possibility of more tie decisions on the output decision variables.

In the case of M=4, L=3 (Figure 7.2-4), receiver performances appear similar to the behavior exhibited by the parameter set M=2, L=3 in that two distinguishable groups are presented. These are the AGC types (better performances) versus the square-law LCR and HD receivers. Throughout most of the strong and moderate jamming regions (2 to 35 dB), it is noticed that the ACJ-AGC performance is up to 2 dB worse than the IC-AGC; this again

being due to the unbalancing effect of the normalization weighting scheme of the ACJ-AGC receiver. The difference between the two AGC performances for M=4 is larger than for M=2 (see Section 7.3 for more discussion of this phenomenon). As for the HD receiver, it proves to be better than the HD cases for L=2 yet exhibits poorer performance than HD for M=2, L=3. Although there are no output decision variable ties for L=3, the HD receiver with more channels will now suffer increased quantization effects in approximating the LCR.

With regard to receiver performances in little thermal noise, Figure 7.2-5 depicts performance results of three candidates (IC-AGC, ACJ-AGC, SNORM) for the parameter set M=2, L=2 at $E_b/N_0 = 20$ dB. These receivers represent the previously shown most ideal performers (AGC types) and the more realizable SNORM receiver. The HD receiver, although a relatively simple ECCM diversity technique in practice, is not included in this comparative set due to the "tie" decision factor when L=2.

We observe for full-band jamming that the IC-AGC and ACJ-AGC receivers yield equivalent performances with the SNORM showing a slightly higher BER. Such behavior for the AGC receivers is to be expected in full-band jamming where a Gaussian channel performance is realized and discrimination against a jammed hop is nonexistent. But as γ becomes less than full-band, it is seen that the AGC receiver curves maintain the same negative slope for increasing E_b/N_j with the ACJ-AGC staying about 1 dB worse than the IC-AGC, this inferior performance being due to the previously described imbalancing effect of the ACJ-AGC normalization mechanism. For the SNORM receiver, we see its performance with respect to the AGC receivers as being: (1) about 0.5 dB

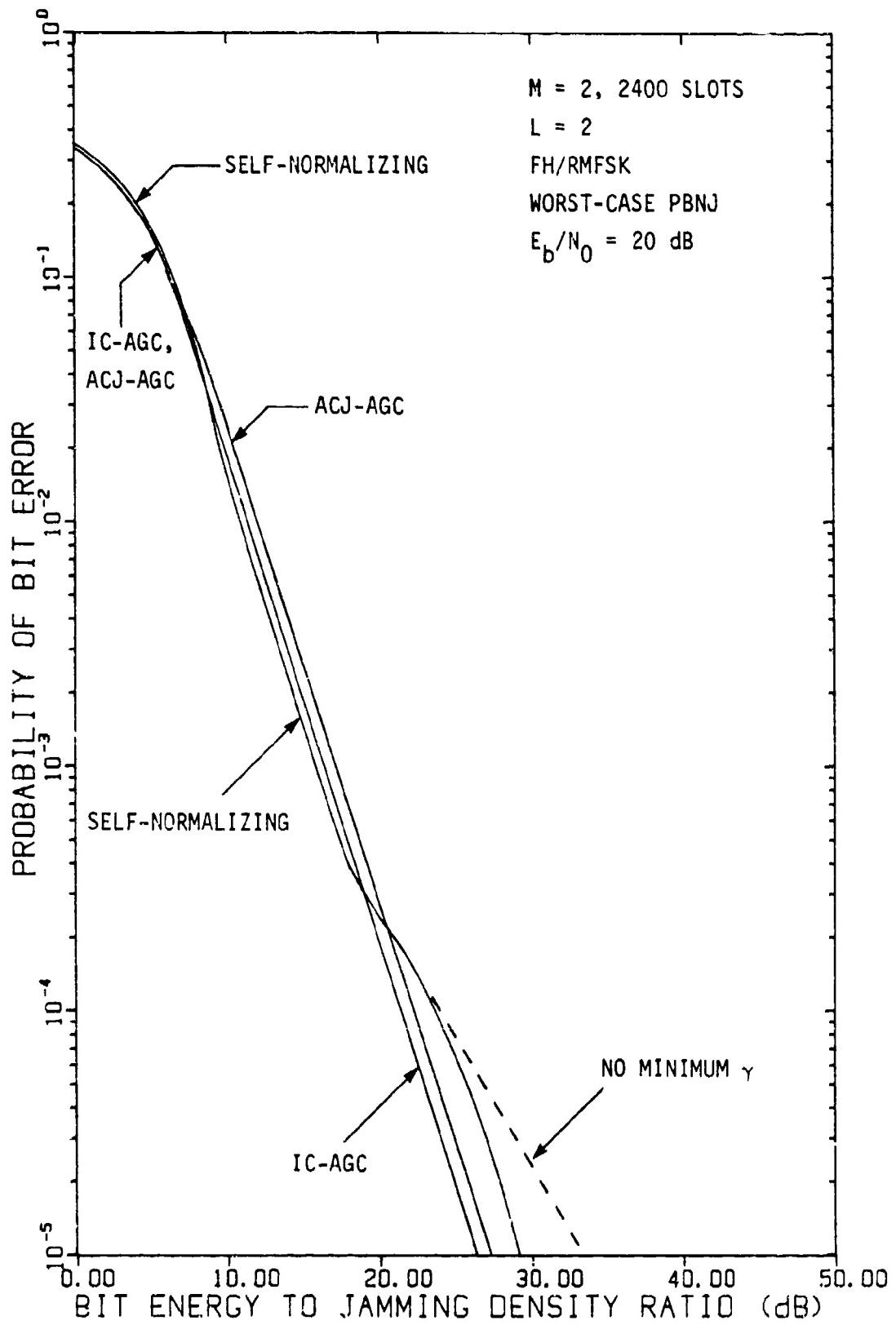


FIGURE 7.2-5 COMPARISON OF DIFFERENT RECEIVERS FOR FH/RMFSK WITH $M=2$ AND $L=2$ HOPS/SYMBOL WHEN $E_b/N_0 = 20$ dB

J. S. LEE ASSOCIATES, INC.

worse in full-band jamming, (2) less than 1 dB better for $\gamma < 1$ (at $E_b/N_J \sim 10$ dB) up until $E_b/N_J > E_b/N_0$, and (3) more than either ACJ-AGC or IC-AGC from about 20 dB to the point where $P(e) = 10^{-5}$ is reached.

An empirical explanation of this phenomenon is obtained by comparing Figure 7.2-6 (IC-AGC) with Figure 7.2-7 (SNORM). These figures show each receiver's individual performance at $E_b/N_0 = 20$ dB for ten different values of γ ranging from $\gamma = 0.001$ to $\gamma = 1.0$ or full-band jamming. We note for the AGC receiver (Figure 7.2-6) that $\gamma = 0.001$ and 0.002 curves produce $P(e) < 10^{-5}$ and thus do not appear on the performance plots. Upon observing the eight remaining γ -curves in these AGC plots, we see each of these $P(e)$ curves contributing the same smooth behavior toward producing an optimum γ -curve result which is a straight line for $\gamma < 1.0$, that is, a slope equal to $A/(E_b/N_J)^2$ where A is some constant defining the inverse-linear relationship existing between γ and available jamming power when $E_b/N_0 = 20$ dB.

However, for the SNORM case we find performance curves for $\gamma = 0.005$ through $\gamma = 0.2$ exhibiting behavior resulting in an optimum γ -curve which is not constant; over these γ ranges the SNORM performance is superior to the IC-AGC receiver. The upper envelope of the curves at first is proportional to $(E_b/N_J)^{-2}$, then transitions to a dependence on $(E_b/N_J)^{-1}$. For infinite E_b/N_0 , from [21] we expect the SNORM and IC-AGC BER's to be proportional to $(E_b/N_J)^{-2}$ indefinitely; for $L=3$ and MFSK the SNORM in [21] is shown to be dependent on $(E_b/N_J)^{-2}$ for no thermal noise, but the AGC is dependent on $(E_b/N_J)^{-3}$, as shown in Figure 7.2-8, taken from [21]. Therefore, the better performance of SNORM for high SNR is not to be expected in all cases of M and L . (See Section 7.3.3.5 for further discussion on the SNORM performance.)

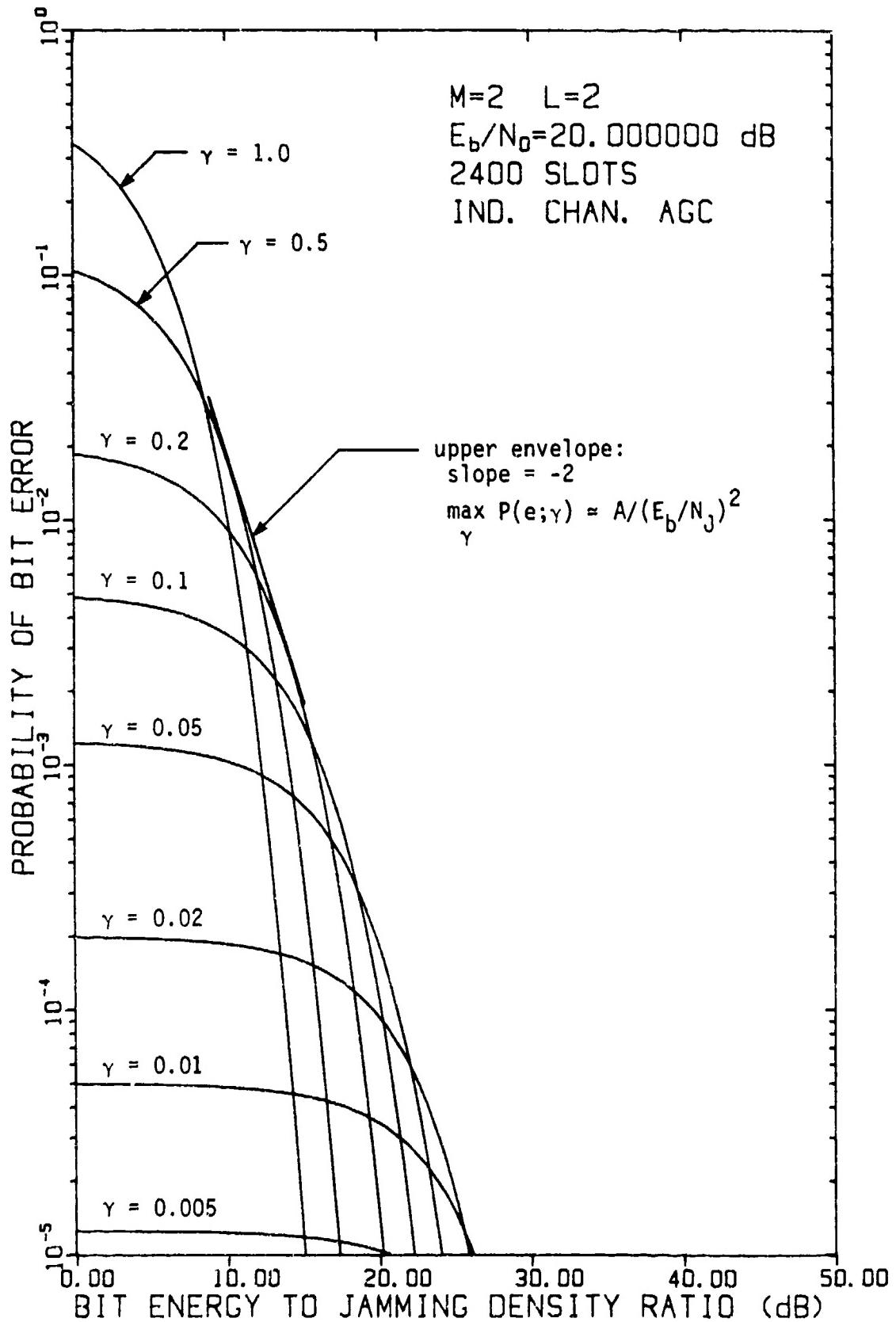


FIGURE 7.2-6 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING NOISE DENSITY RATIO FOR IC-AGC RECEIVER IN WORST-CASE PARTIAL-BAND NOISE FOR $M=2, L=2$ AT $E_b/N_0 = 20 \text{ dB}$

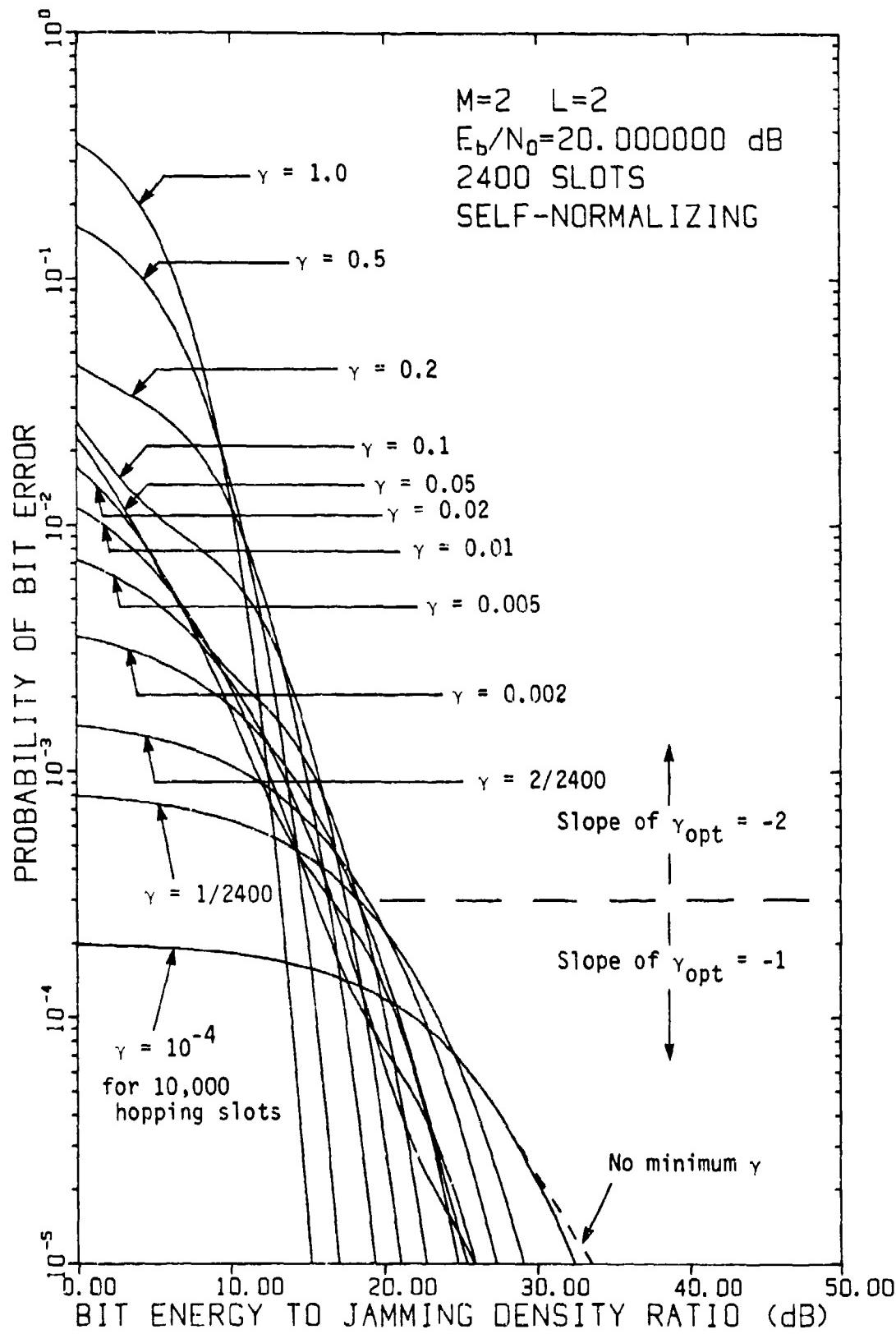


FIGURE 7.2-7 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING NOISE DENSITY RATIO FOR SELF-NORMALIZING RECEIVER IN WORST-CASE PARTIAL-BAND NOISE JAMMING FOR $M=2$, $L=2$ AT $E_b/N_0 = 20 \text{ dB}$

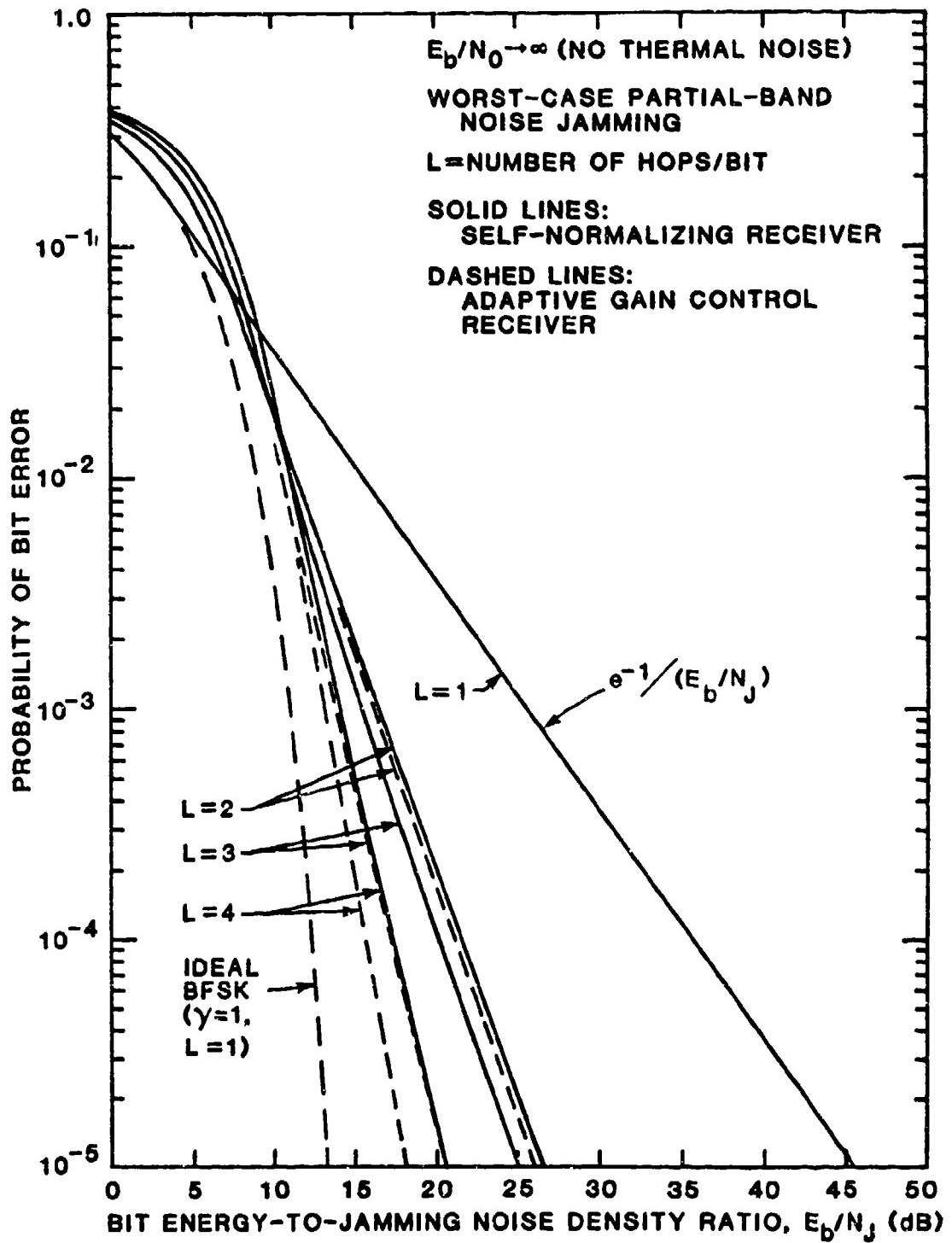


FIGURE 7.2-8 WORST-CASE PARTIAL-BAND NOISE JAMMING PERFORMANCE OF THE SELF-NORMALIZING FH/BFSK RECEIVER WHEN THERMAL NOISE IS ABSENT, WITH THE NUMBER OF HOPS/BIT (L) AS A PARAMETER

J. S. LEE ASSOCIATES, INC.

7.3 COMPARISONS OF FH/RMFSK AND FH/MFSK

Having compared the performances of the various FH/RMFSK receivers in worst-case partial-band noise jamming (WCPBNJ), we now consider the differences in performance to be expected between the random hopping MFSK system studied in this report (FH/RMFSK) and the conventional (adjacent or contiguous) hopping MFSK system (FH/MFSK) studied, for example, in [1].

7.3.1 The M=2, L=1 Case

It was found by Blanchard [6] that for M=2 and L=1 the two systems yield the same performance, at least for the $E_b/N_0 = 30$ dB case he studied, with the differences in possible jamming events accounted for by different optimum values of γ , the fraction of the system bandwidth which is jammed. (Typically, for high E_b/N_j the RMFSK γ_{opt} was found to be half that for MFSK.) Figure 7.3-1 displays the cases Blanchard considered, except that we use $E_b/N_0 = 13.35$ dB, corresponding to a 10^{-5} BER with no jamming. Our results indicate that the two systems do indeed perform the same for M=2 and L=1, except for certain differences for weak jamming (high E_b/N_j). What is the significance of the differences we observe in these computed results?

Curve A in Figure 7.3-1 is from [1] and represents the quantity

$$\max_{0 < \gamma \leq 1} \left[\frac{1}{2}(1-\gamma) e^{-E_b/2N_0} + \frac{1}{2}\gamma e^{-E_b/2N_T} \right], \quad (7.3-1a)$$

where

$$\frac{E_b}{N_T} = \frac{\frac{E_b}{N_0} \cdot \frac{\gamma E_b}{N_J}}{\frac{E_b}{N_0} + \frac{\gamma E_b}{N_J}}. \quad (7.3-1b)$$

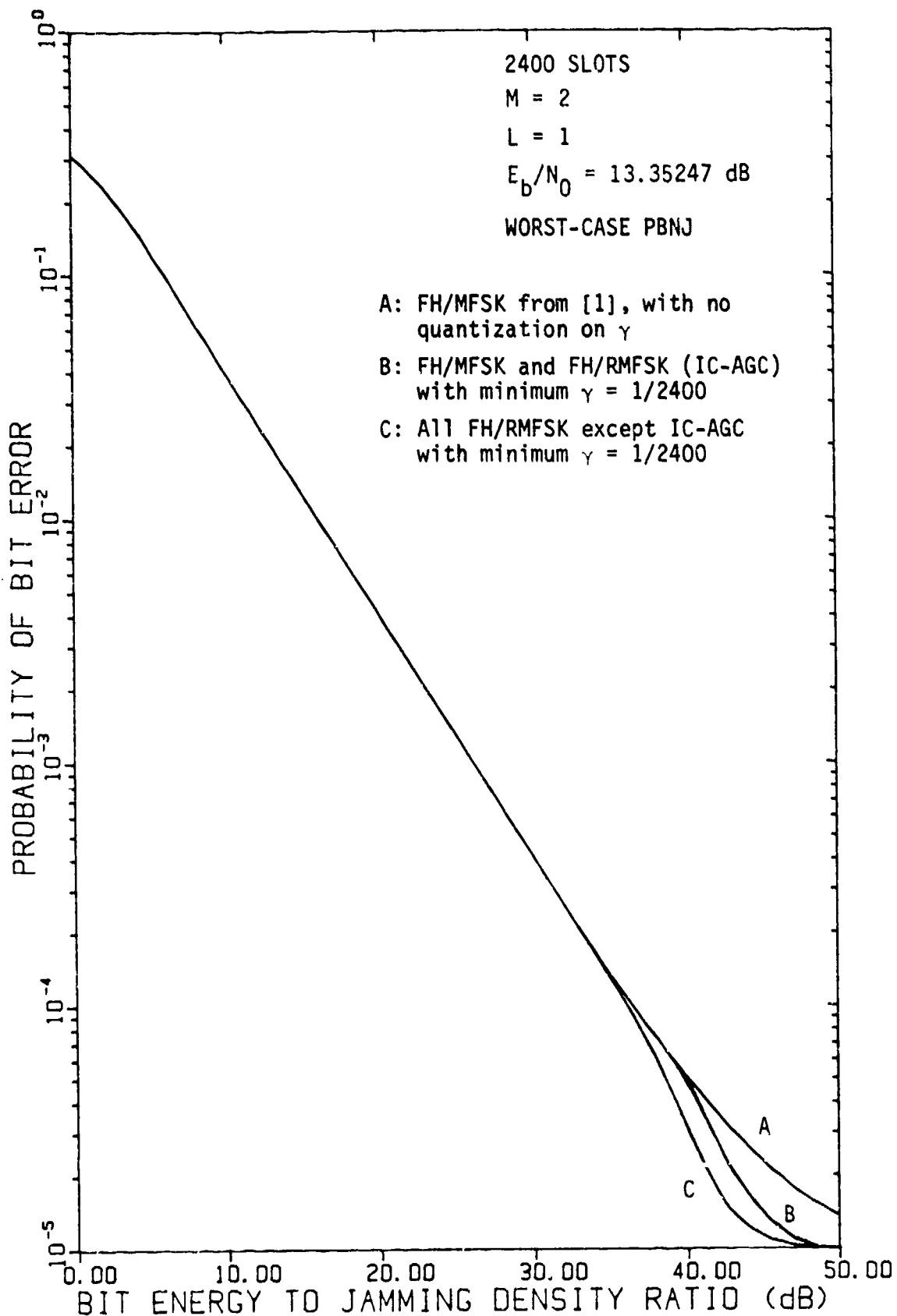


FIGURE 7.3-1 COMPARISON OF PERFORMANCE OF ALL RECEIVERS FOR FH/MFSK AND FH/RMFSK WITH M=2 AND L=1 WHEN $E_b/N_0 = 13.35247 \text{ dB}$

J. S. LEE ASSOCIATES, INC.

In this formulation for the binary case it is assumed that both slots are either jammed (with probability γ) or unjammed (with probability $1-\gamma$).

Curve B represents the quantity

$$\max_{1 \leq q \leq N} \left[\frac{1}{2} p_0 e^{-E_b/2N_0} + \frac{1}{2} p_1 e^{-E_b/2N_T} \right], \quad (7.3-2a)$$

where

$$p_1 = 1 - q/N = 1 - p_0. \quad (7.3-2b)$$

This formulation assumes that $\gamma = q/N$ is quantized - a discrete number (q) of the total number slots (N) are jammed, with the minimum γ equal to $1/N$. It further assumes that the system is FH/RMFSK with IC-AGC processing; a more explicit form of the error expression is

$$P(e;q) = (\pi_0 + \pi_1) \cdot \frac{1}{2} e^{-E_b/2N_0} + (\pi_1 + \pi_2) \cdot \frac{1}{2} e^{-E_b/2N_T}, \quad (7.3-3)$$

where

$$\pi_r = \text{prob. that } r \text{ slots are jammed}, \quad (7.3-4a)$$

and

$$\pi_0 = \frac{N-q}{N} \cdot \frac{N-q-1}{N-1} \quad (7.3-4b)$$

$$\pi_1 = \frac{N-q}{N} \cdot \frac{q}{N-1} \quad (7.3-4c)$$

$$\pi_2 = \frac{q}{N} \cdot \frac{q-1}{N-1}. \quad (7.3-4d)$$

Because of the individual channel normalization, the conditional BER depends only on whether the signal channel is jammed. Thus here are only two terms, with weights $p_0 = \pi_0 + \pi_1$ and $p_1 = \pi_1 + \pi_2$.

J. S. LEE ASSOCIATES, INC.

Now, the only difference between (7.3-2) and (7.3-1) is the quantization and minimum value of $\gamma = \frac{q}{N}$. Therefore in Figure 7.3-1 we identify curve B also with binary FH/MFSK, even though $q = 1$ violates the assumption that both channels are together jammed or unjammed.

Curve C in Figure 7.3-1 represents the quantity

$$\max_{1 \leq q \leq N} \left[\pi_0 \cdot \frac{1}{2} e^{-E_b/2N_0} + \pi_1 \cdot e^{-(E_b/N_0)/(K+1)} + \pi_2 \cdot \frac{1}{2} e^{-E_b/2N_T} \right], \quad (7.3-5a)$$

where

$$K = \sigma_T^2/\sigma_N^2 = (E_b/N_0)/(E_b/N_T). \quad (7.3-5b)$$

This is the BER for all the FH/RMFSK receivers except the IC-AGC, and allows for only one of the two channels to be jammed.

Thus, in general, our results agree with Blanchard's conclusion that FH/RMFSK performs the same as FH/MFSK for $M=2$ and $L=1$, neglecting small asymptotic differences connected with assumptions on the quantization and minimum value of γ . Now we consider whether his conjecture that the two hopping systems perform the same for $M>2$ and $L=1$ is correct, and how the comparison is affected by $L>1$. In what follows, we shall use the fact that the IC-AGC FH/RMFSK receiver performs essentially the same as the AGC FH/MFSK receiver.

7.3.2 $L=1$ with Alphabet Size Varied.

In order to compare RMFSK and MFSK for $L=1$ and $M>1$, it is sufficient to consider Figures 7.3-2 and 7.3-3.

In Figure 7.3-2 the performances of the FH/MFSK and the IC-AGC FH/RMFSK receivers are shown for $L=1$ and $M=2,4,8$. The values of E_b/N_0 used were chosen

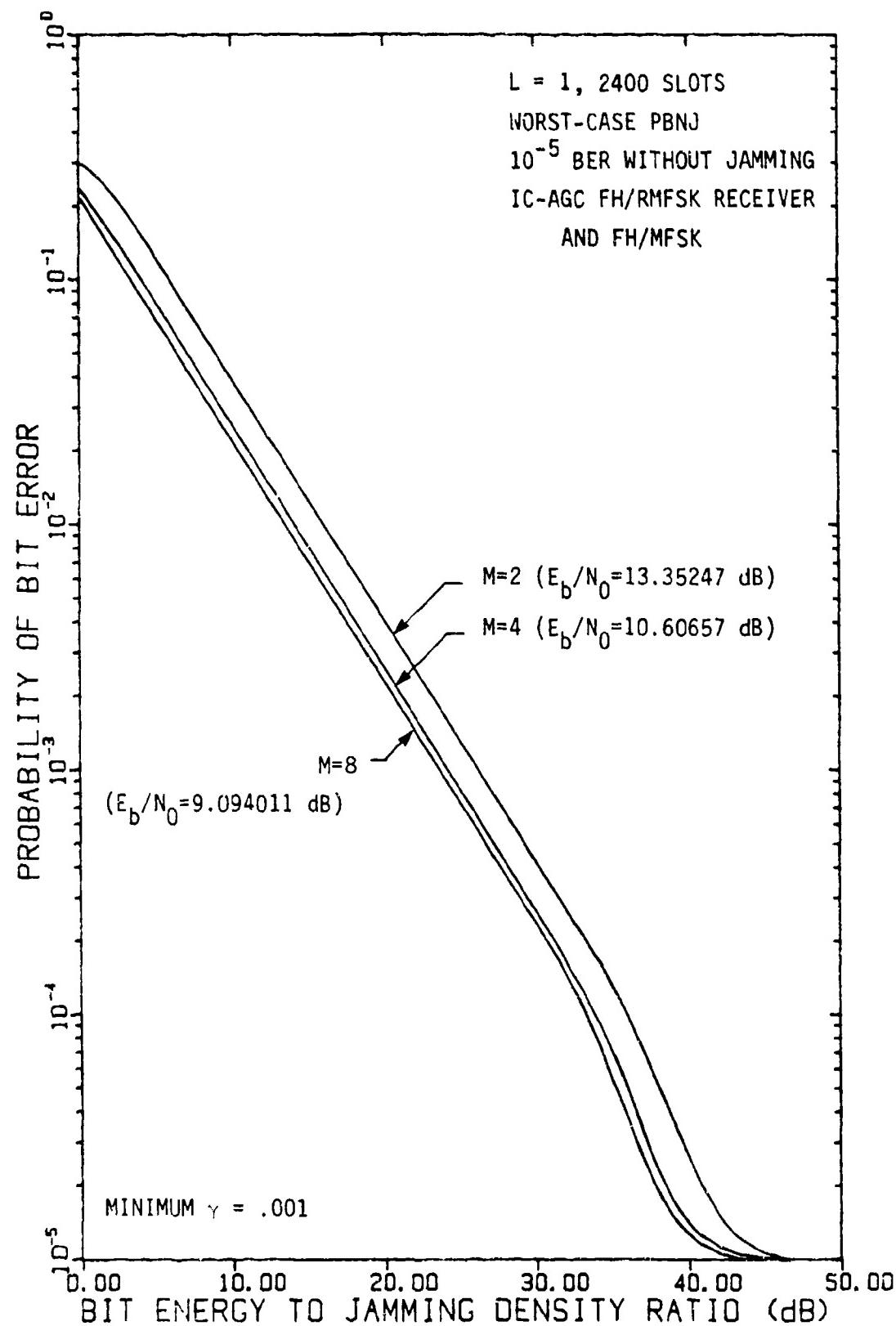


FIGURE 7.3-2 PERFORMANCE OF INDIVIDUAL CHANNEL AGC RECEIVER FOR FH/RMFSK WHEN $L=1$ HOP/SYMBOL WITH M AS A PARAMETER AND E_b/N_0 CORRESPONDING TO $P_b(e) = 10^{-5}$ IN THE ABSENCE OF JAMMING.

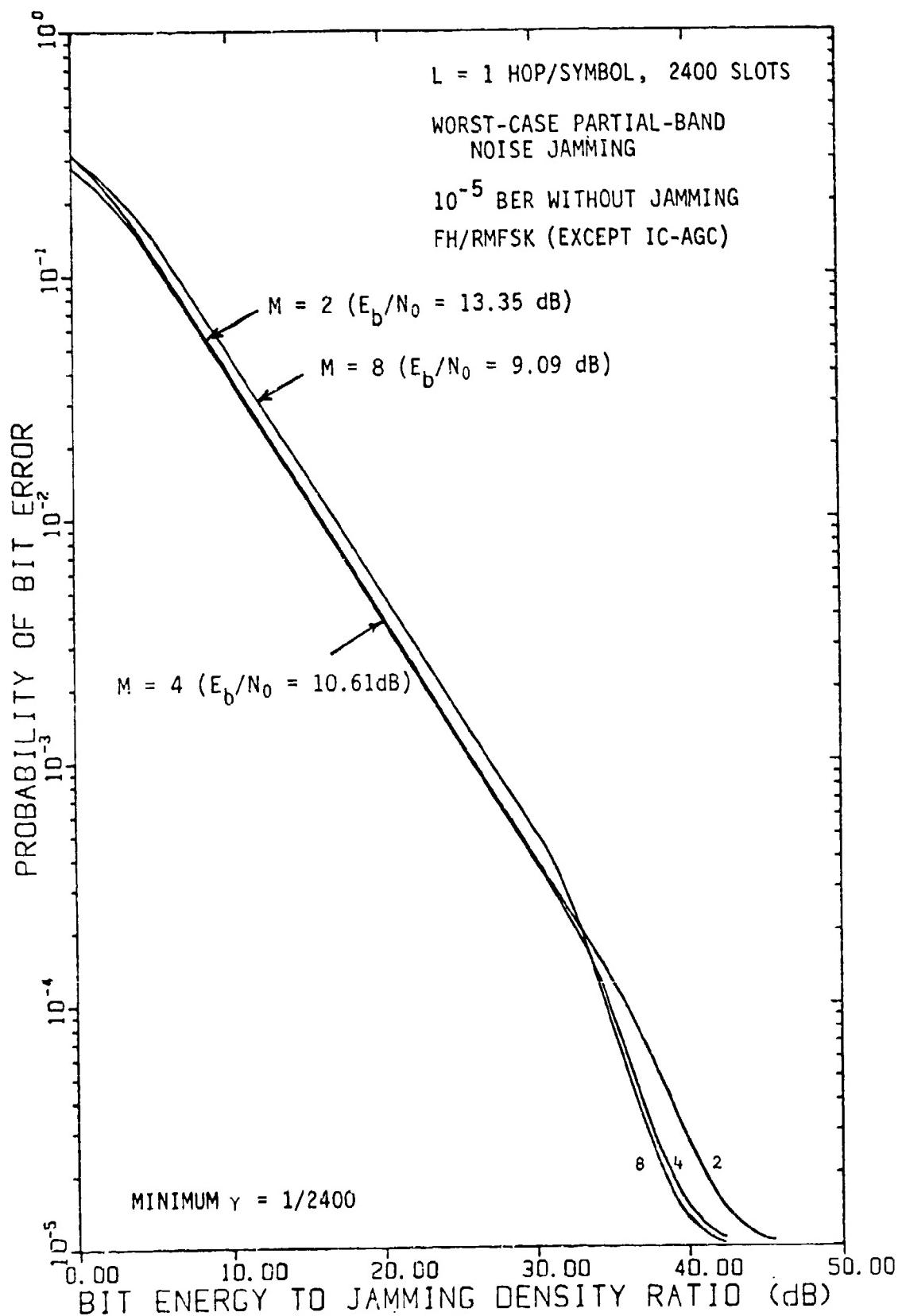


FIGURE 7.3-3 WORST-CASE PARTIAL-BAND NOISE JAMMING PERFORMANCE OF FH/RMFSK RECEIVERS FOR $L = 1$ HOP/SYMBOL AND $M = 2, 4, 8$ WHEN E_b/N_0 GIVES A 10^{-5} BER WITHOUT JAMMING

J. S. LEE ASSOCIATES, INC.

to give each example a 10^{-5} BER under no jamming. We observe from these results that for these systems the BER decreases as the alphabet size M increases, the conventional interpretation of which is that an "M-ary coding gain" is at work. This is the phenomenon usually observed for MFSK systems in the Gaussian interference channel.

In Figure 7.3-3 the same parameters are used as in Figure 7.3-2, but now the receivers are the FH/RMFSK receivers (except IC-AGC), which have identical performance for $L=1$. For these receivers we find that for strong jamming the system performance does not consistently improve as M increases, but instead improves very slightly for $M=4$ and degrades for $M=8$. Clearly this is the result of the increased probability, as M increases, of the most damaging jamming event: jamming power in a non-signal slot but not in the signal slot. Since the $M=2$ performances in the two figures are virtually the same, we conclude that FH/RMFSK is consistently more vulnerable to WCPBNJ than is FH/MFSK for $M>2$. The difference is about 3 dB for $M=4$ and 5 to 6 dB for $M=8$.

When the jamming is weak, we expect the relative performances for different M to approach the usual non-jammed behavior, and this is observable in Figure 7.3-3 for $E_b/N_j > 34$ dB.

7.3.3 Cases Where $L>1$ Hop/Symbol

Since the various FH/RMFSK receivers and their FH/MFSK counterparts begin to exhibit different performances when diversity is used ($L>1$), it is necessary to consider them separately. Most of the FH/MFSK results are taken

J. S. LEE ASSOCIATES, INC.

from [1]; however, when convenient we shall continue to utilize the fact that IC-AGC FH/RMFSK receiver performs essentially the same as the AGC FH/MFSK receiver.

7.3.3.1 Linear Combining Receiver.

We begin by comparing the performance of the square-law linear combining receiver for both FH/RMFSK and RH/MFSK signalling strategies. Figures 7.3-4 and 7.3-5 show these performances for $M=2$, $L=2$ and $M=4$, $L=2$ respectively. In both figures, it is apparent that MFSK is superior to RMFSK, ignoring the effects of the different minimum γ value used in the computations. This vulnerability of RMFSK can be attributed to what we term the "unbalancing" error mechanism inherent in partial-band jamming of RMFSK. Specifically, the random placement of M -ary slots over the hopping bandwidth W allows more chance for a jamming hit than does a block-hopping MFSK signal where it is assumed that either all M slots will be jammed or unjammed. This probability increases for greater values of M as evidenced by Figure 7.3-5.

7.3.3.2 AGC receivers.

Comparisons for $L>1$ among the AGC-type nonlinear combining receivers are exhibited in Figures 7.3-6 to 7.3-9. Recalling that RMFSK and MFSK hopping systems perform virtually the same for $M=2$ and $L=1$, it is instructive to observe in Figures 7.3-6 and 7.3-7 that the system performances differ by about 1 dB when $L=2$ or $L=3$. The approximately 3 dB difference noted for $M=4$ and $L=1$ continues to hold for $M=4$ and $L=2$ or 3, as shown in Figures 7.3-8 and 7.3-9.

7.3.3.3 Clipper receiver.

A very interesting consideration is brought to light by Figure 7.3-1 which shows the clipper receiver's FH/RMFSK performance for $L=2$ and several values of M . It was found analytically in Section 5 that the optimum clipping

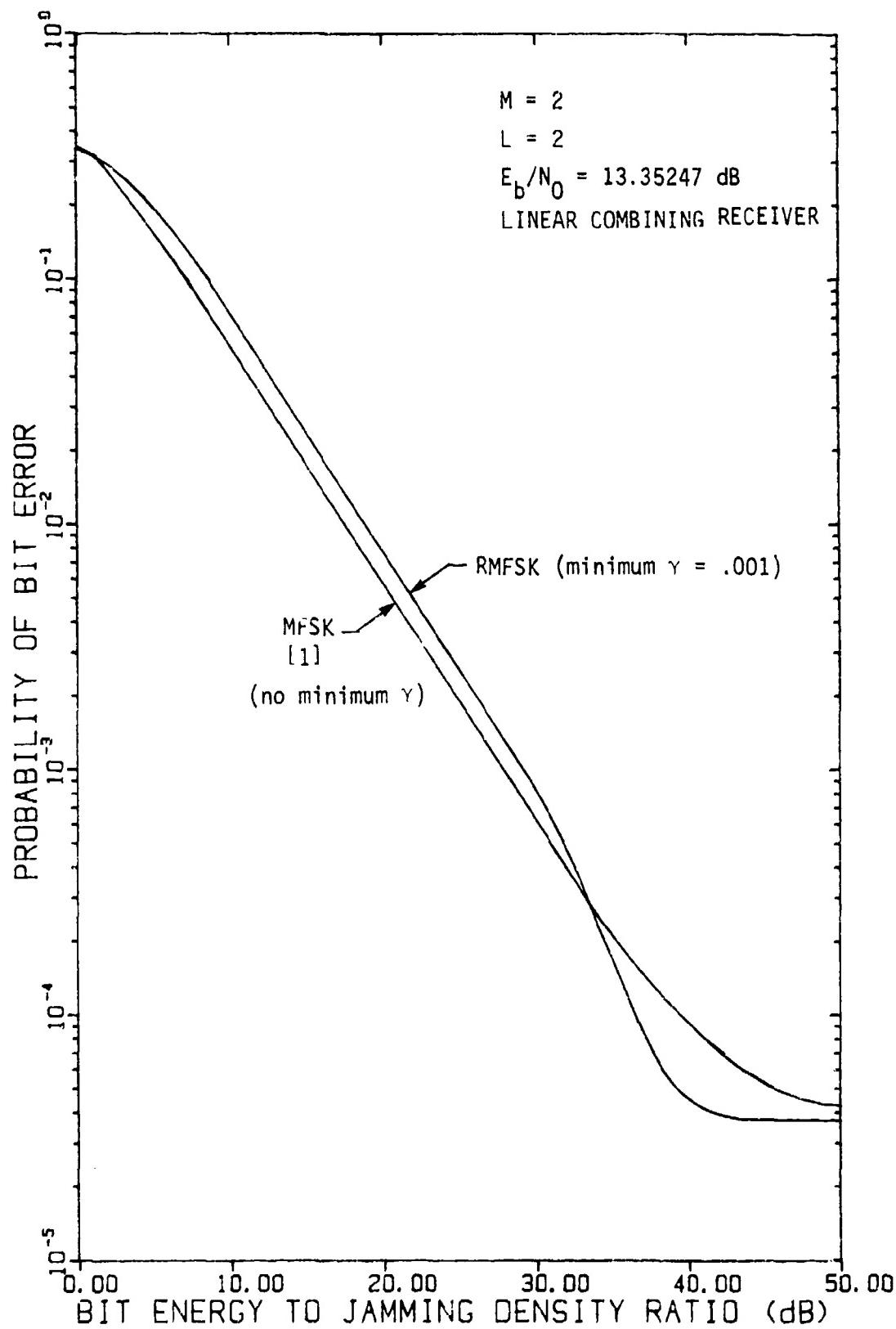


FIGURE 7.3-4 COMPARISON OF SQUARE-LAW LINEAR COMBINING RECEIVERS FOR FH/MFSK AND FH/RMFSK WITH $M=2$ AND $L=2$ HOPS/SYMBOL WHEN $E_b/N_0 = 13.35247 \text{ dB}$

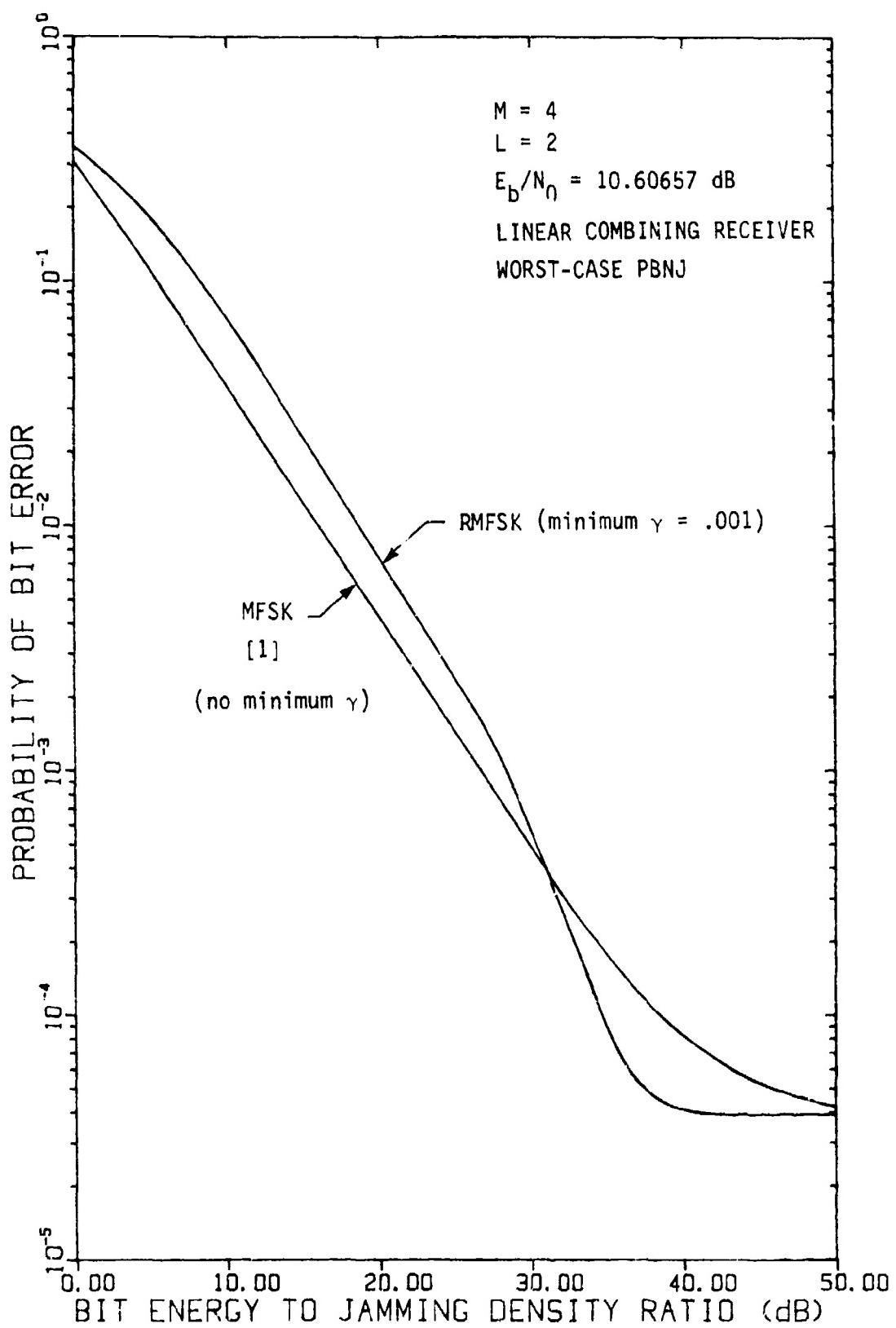


FIGURE 7.3-5 COMPARISON OF SQUARE-LAW LINEAR COMBINING RECEIVERS FOR FH/MFSK AND FH/RMFSK WITH $M=4$ AND $L=2$ HOPS/SYMBOL WHEN $E_b/N_0 = 10.60657 \text{ dB}$

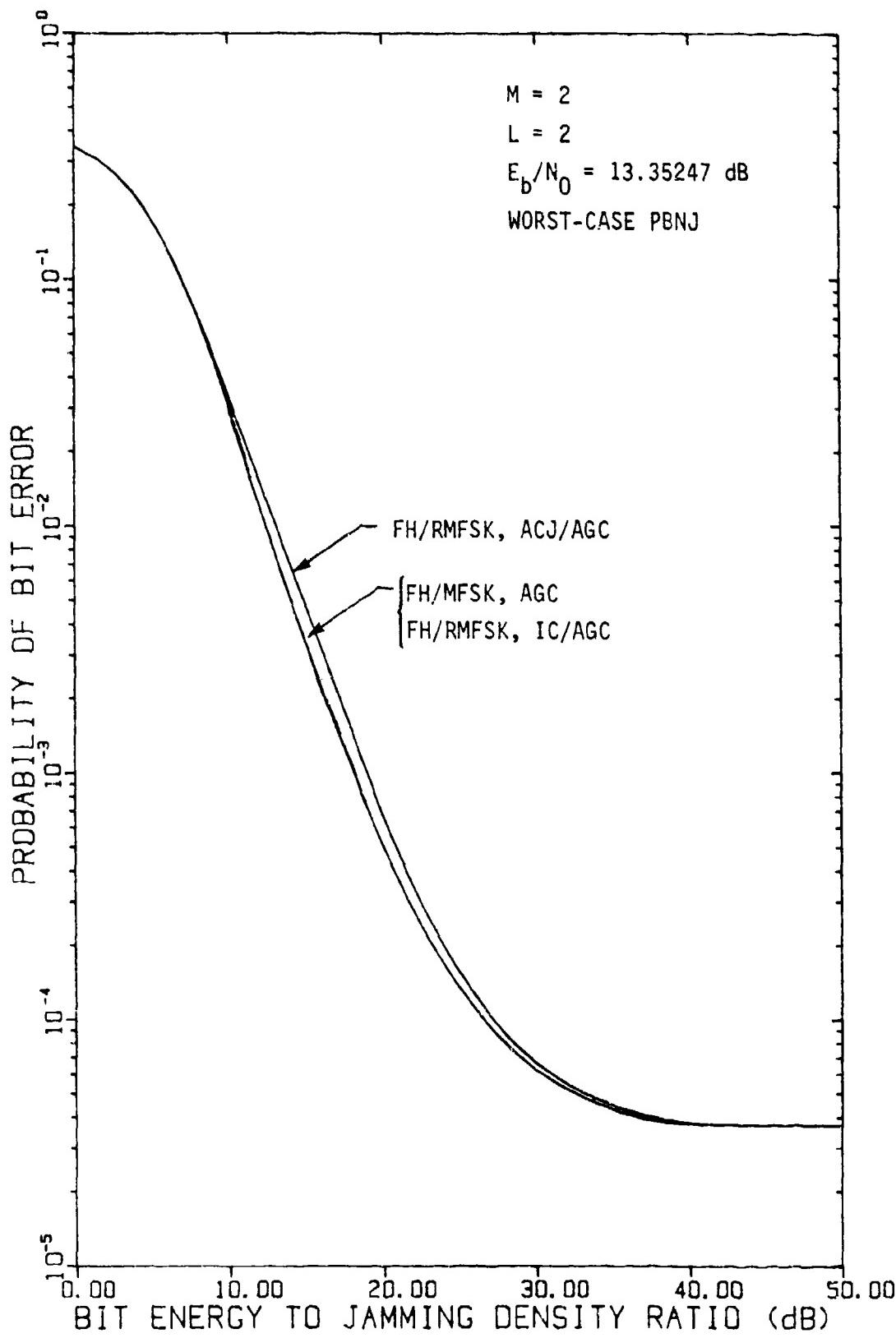


FIGURE 7.3-6 COMPARISON OF AGC RECEIVERS FOR FH/MFSK AND FH/RMFSK WITH $M=2$ AND $L=2$ HOPS/SYMBOL WHEN $E_b/N_0 = 13.35247 \text{ dB}$

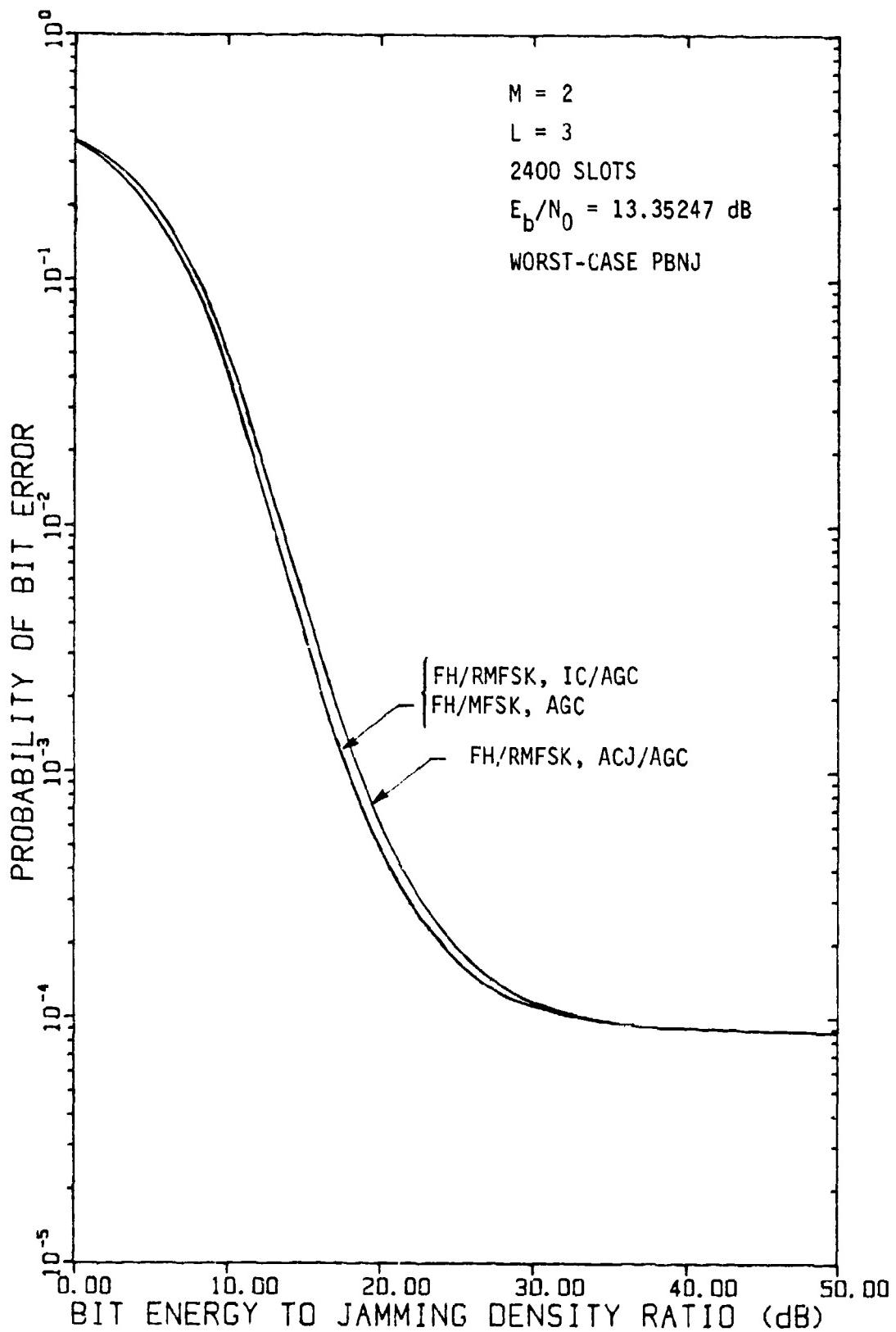


FIGURE 7.3-7 COMPARISON OF AGC RECEIVERS FOR FH/MFSK AND FH/RMFSK WITH $M=2$ AND $L=3$ HOPS/SYMBOL WHEN $E_b/N_0 = 13.35247 \text{ dB}$

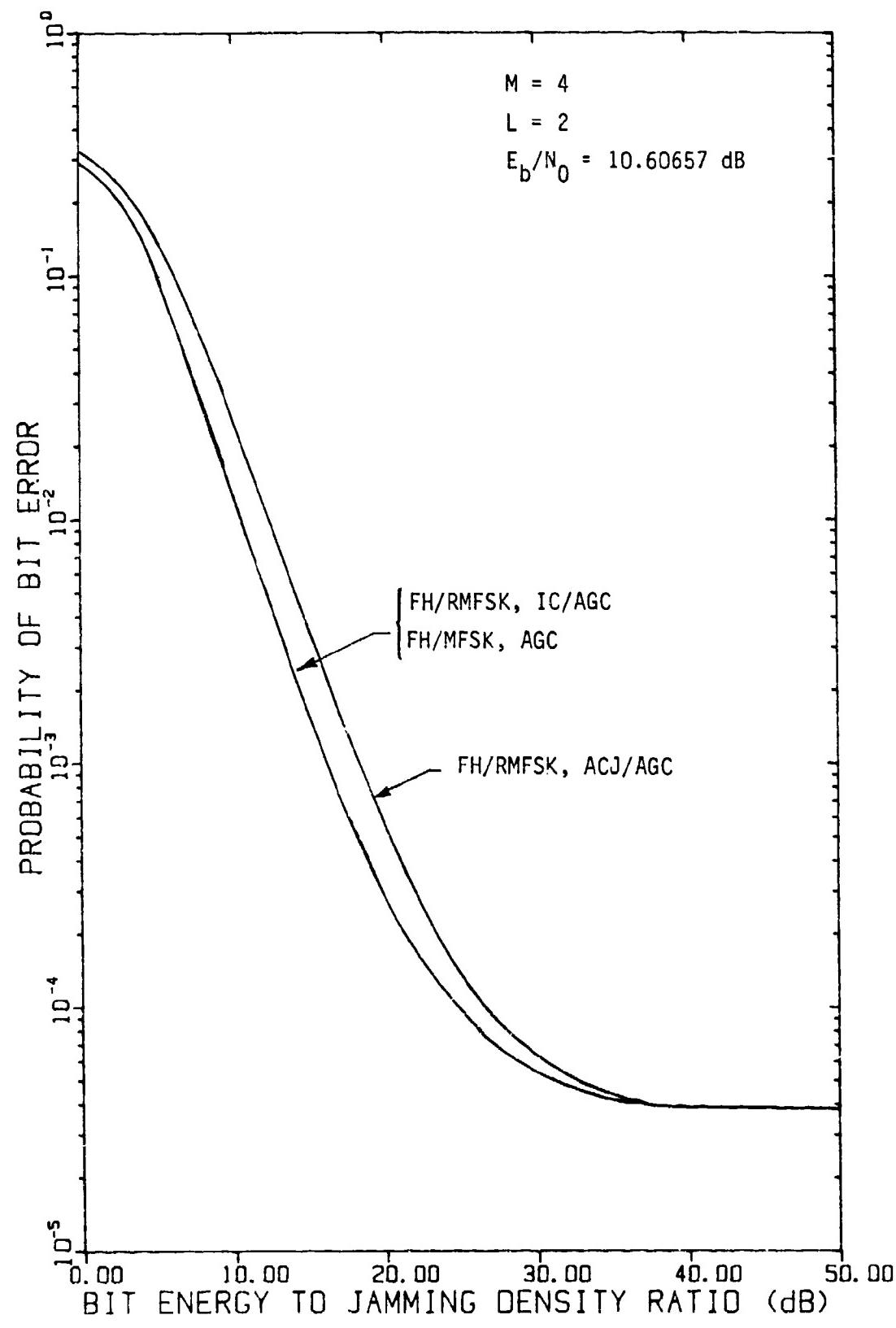


FIGURE 7.3-8 COMPARISON OF AGC RECEIVERS FOR FH/MFSK AND FH/RMFSK WITH $M=4$ AND $L=2$ HOPS/SYMBOL WHEN $E_b/N_0 = 10.60657 \text{ dB}$

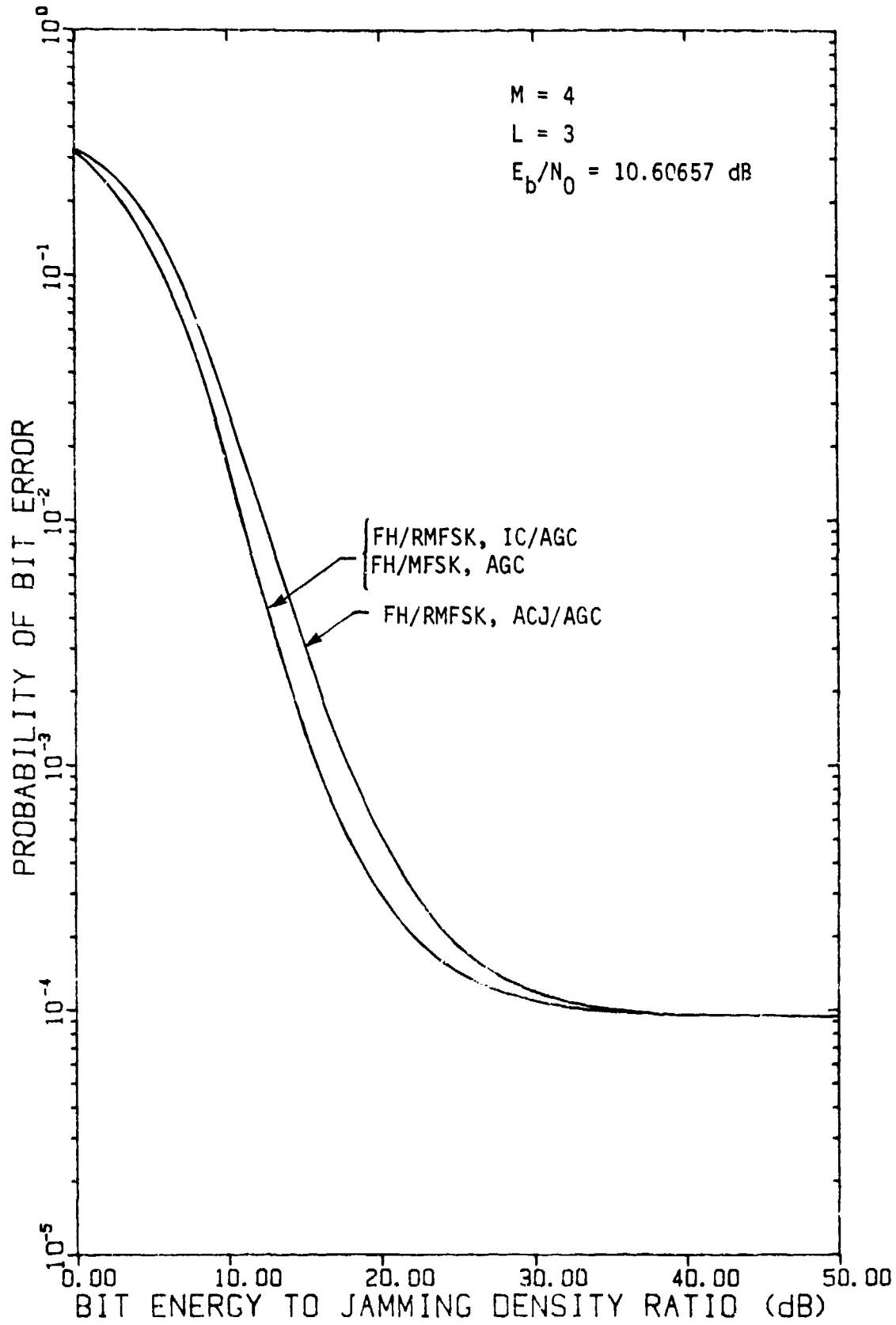


FIGURE 7.3-9 COMPARISON OF AGC RECEIVERS FOR FH/MFSK AND FH/RMFSK WITH $M=4$ AND $L=3$ HOPS/SYMBOL WHEN $E_b/N_0 = 10.60657 \text{ dB}$

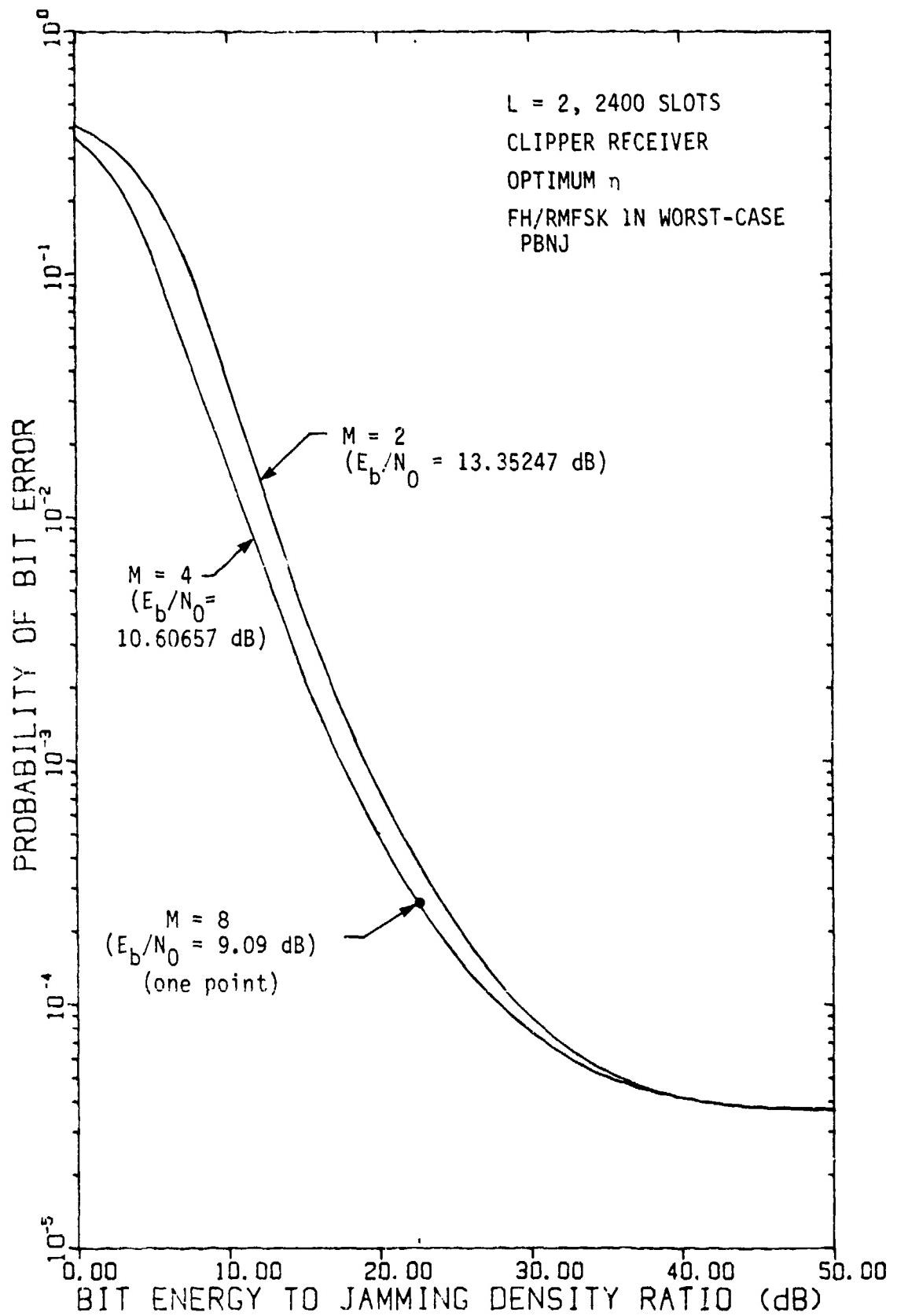


FIGURE 7.3-10 PERFORMANCE OF CLIPPER RECEIVER FOR FH/RMFSK WHEN $L=2$ HOPS/SYMBOL WITH M AS A PARAMETER AND E_b/N_0 CORRESPONDING TO $P_b(e) = 10^{-5}$ IN THE ABSENCE OF JAMMING (WHEN $L=1$ HOP/SYMBOL)

J. S. LEE ASSOCIATES, INC.

threshold for $L=1$ is infinite (no clipping), whereas numerically it was determined that a finite threshold is optimum for $L>1$. Consequently, the $L=1$ "clipper" receiver is not a clipping receiver at all but one identical in operation to all the other RMFSK receivers for $L=1$ except IC-AGC, and its FH/RMFSK performance tends to get worse for increasing M as demonstrated previously in Figure 7.3.3. However, for $L=2$ we observe from Figure 7.3-10 that the clipper is performing in a manner similar to the IC-AGC, in that increasing M from 2 to 4 reduces the BER; however further increase to $M=8$ degrades performance. The reason for this similarity in behavior is that the clipper receiver, like the IC-AGC, operates to limit jamming input to the soft decision on an individual channel basis. The clipper is in this sense a crude version of the IC-AGC; but for higher values of M the losses become significant and the performance trend resembles the other RMFSK receivers more than the IC-AGC receiver. We then would expect clipping to be advantageous against jamming for $L=1$ as well; but the threshold was optimized for no jamming in order to avoid requiring the receiver to know or measure jamming parameters. If the threshold were jamming-dependent, the clipper receiver might follow the IC-AGC more closely for higher M . The tendency of the clipper receiver to "emulate" the IC-AGC was observed earlier in Figure 7.2-3, where we see that this tendency is more pronounced for strong jamming.

7.3.3.4 Hard-decision receiver.

Now if the clipper receiver can be thought of as a crude version of the IC-AGC, the hard-decision (HD) receiver can be considered a crude version of the ACJ-AGC because both act to limit or de-emphasize the entire set of M channels on a jammed hop, rather than operating on the channels separately.

J. S. LEE ASSOCIATES, INC.

Thus we observe in Figure 7.3-11 the tendency for the HD receiver's BER to increase with M (after M>4) in strong jamming, just like the ACJ-AGC receiver's BER, and thereby to yield a worse performance for RMFSK than for MFSK. In weak or no jamming, the HD's BER for L>1 gets worse for increasing M (unlike the other, soft-decision receivers) because noncoherent combining losses are in effect amplified by the quantization the HD uses.

7.3.3.5 Self-normalizing receiver.

In the previous examples, we have observed a consistent trend for RMFSK hopping to yield no better--and sometimes worse--performance than conventional MFSK hopping. This was explained as being due to the possibility of jamming being present in a non-signal channel but not in the signal channel for RMFSK but not for MFSK. It is also true that using RMFSK there can be jamming only in the signal channel, which tends to favor a correct decision. Apparently, using the LCR, AGC, clipper, and HD receivers, the jamming of one channel has a net effect of degrading the system performance for L>1.

Now, we consider the comparison of RMFSK with MFSK using the self-normalizing receiver, and will see an exception to the trend previously observed. Figure 7.3-12 displays the SNORM error performances for FH/RMFSK and RH/MFSK in WCPBNJ for M=L=2 and $E_b/N_0 = 13.35$ dB and 20 dB. We see that the BER for RMFSK is better than for MFSK. This behavior seems to be connected with the jamming events in which both channels are either jammed or unjammed on a given hop, rather than those for which only one channel is jammed. This statement is supported by the fact that (1) the curves are roughly parallel for moderate-to-weak jamming (the portion of

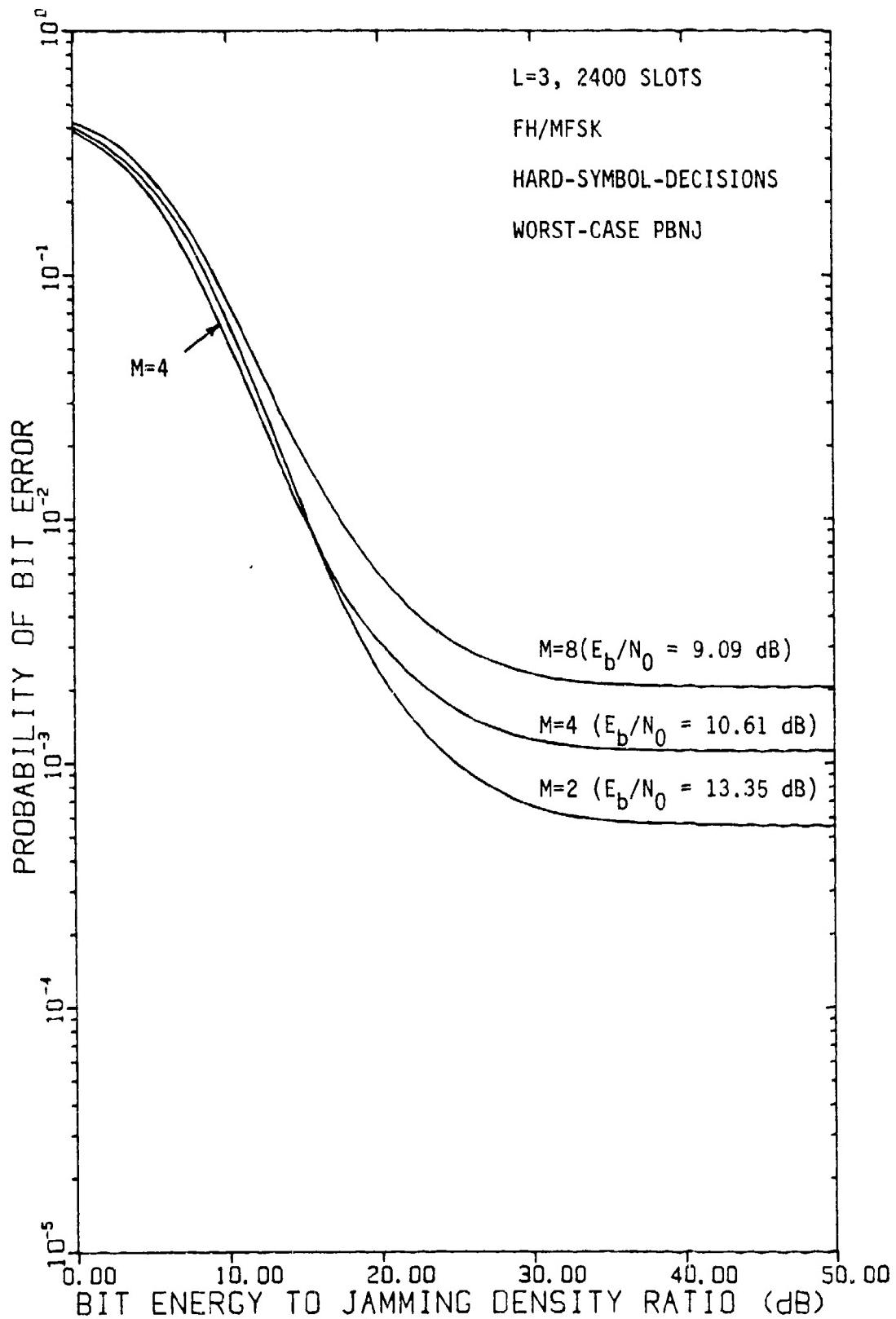


FIGURE 7.3-11 ERROR PERFORMANCE FOR HARD SYMBOL DECISION FH/RMFSK RECEIVER IN WORST-CASE PARTIAL-BAND NOISE JAMMING WHEN $L=3$ HOPS/SYMBOL; $M=2,4,8$; AND E_b/N_0 GIVES 10^{-5} ERROR WITHOUT JAMMING WHEN $L=1$

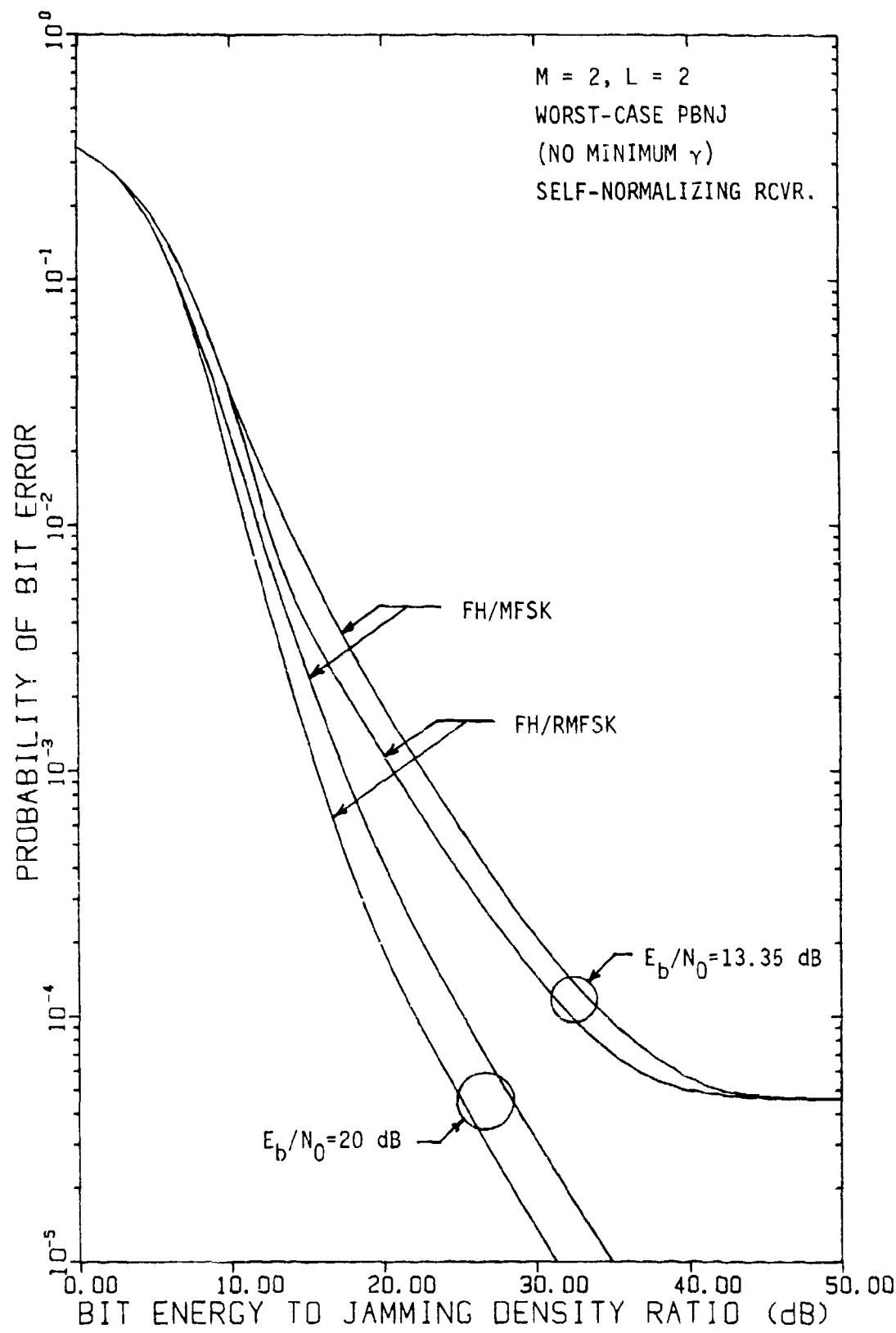


FIGURE 7.3-12 WORST-CASE PARTIAL-BAND NOISE JAMMING PERFORMANCES OF FH/RMFSK AND FH/MFSK USING THE SELF-NORMALIZING RECEIVER, FOR $M=2$ AND $L=2$

J. S. LEE ASSOCIATES, INC.

the curve proportional to e^{-E_b/N_J}), and (2) the MFSK receiver is subject only to those particular jamming events, by assumption. The advantage of RMFSK, according to this interpretation, then lies in the smaller probability of both channels being jammed on a given hop.

Now if M were increased to $M=4$ or $M=8$, it would be expected that the effects of jamming in one channel only would tend to increase the RMFSK error, since then a damaging effect would be $M-1$ times as likely as a helping effect.

On the other hand, the improvement of the RMFSK error over that of MFSK can be explained in terms of how the SNORM receiver processes the jamming events for which only one channel is jammed, in contrast to the way the other RMFSK receivers process the events. When only the non-signal channel is jammed the hop statistics are ($K = \sigma_T^2/\sigma_N^2 \gg 1$)

$$z_{1k} = \frac{\chi^2(2, 2\rho_N)}{\chi^2(2, 2\rho_N) + K\chi^2(2)} \rightarrow 0 \quad (7.3-6a)$$

$$z_{2k} = \frac{K\chi^2(2)}{\chi^2(2, 2\rho_N) + K\chi^2(2)} \rightarrow 1. \quad (7.3-6b)$$

But when only the signal channel is jammed,

$$z_{1k} = \frac{K\chi^2(2, 2\rho_T)}{K\chi^2(2, 2\rho_T) + \chi^2(2)} \rightarrow 1 \quad (7.3-7a)$$

$$z_{2k} = \frac{\chi^2(2)}{K\chi^2(2, 2\rho_T) + \chi^2(2)} \rightarrow 0. \quad (7.3-7b)$$

That is, the SNORM per-hop processing is nearly equivalent to a hard symbol decision; the signal channel suppresses the nonsignal channel when the signal is jammed, but if the nonsignal channel is jammed, it is awarded a value of at most 1. Thus the receiver de-emphasizes jammed hops while at the same time distinguishing between "good" and "bad" jammed hops. The

J. S. LEE ASSOCIATES, INC.

IC-AGC receiver, by contrast, penalizes the channel being jammed by normalizing its noise to the same variance as the other channel; this has the effect of suppressing the jammed channel when only one is jammed. This is a good thing to do when the non-signal channel is jammed; but it is not beneficial when the signal channel is jammed.

According to this second interpretation of the results the SNORM performance for RMFSK is better because it makes good use of the "favorable" jamming events, which do not occur for MFSK. However, we still would expect the RMFSK performance to degrade for higher M under this interpretation.

7.4

COMPARISON OF RECEIVER DIVERSITY EFFECTS

The FH/RMFSK receivers we have studied are distinguished by their methods of combining the L hops transmitted per MFSK symbol. The objective of the diversity transmission is to spread the signal on a symbol basis, making it less likely that the symbol is jammed for the entirety of its duration. The L pieces of the symbol transmission are then sequentially acquired noncoherently and accumulated after weighting or otherwise processing them individually. Since the combining is done noncoherently, the performance of the system without jamming or with full-band jamming (Gaussian channel) necessarily is degraded from that using one hop with same signal energy. However, when the system bandwidth is jammed partially, giving rise to a type of non-Gaussian interference channel, the system performance is improved using diversity, provided that the hop processing in some fashion limits or discriminates against those hops which are jammed.

The conventional diversity receiver for MFSK, which we have termed the linear combining square-law receiver (LCR), is known to be effective against signal fading, that is, when there exists a random-amplitude signal in a Gaussian channel. But against partial-band noise jamming (PBNJ), the LCR is not effective since jammed hops are not de-emphasized. Figure 7.4-1 illustrates for $M=2$ that LCR performance degrades in proportion to L .

One view of the individual-channel adaptive gain control receiver (IC-AGC), which normalizes each square-law detector sample by its a priori noise variance, is that it in effect renders the non-Gaussian PBNJ interference into a Gaussian interference with unit variance in each MFSK slot. The residual

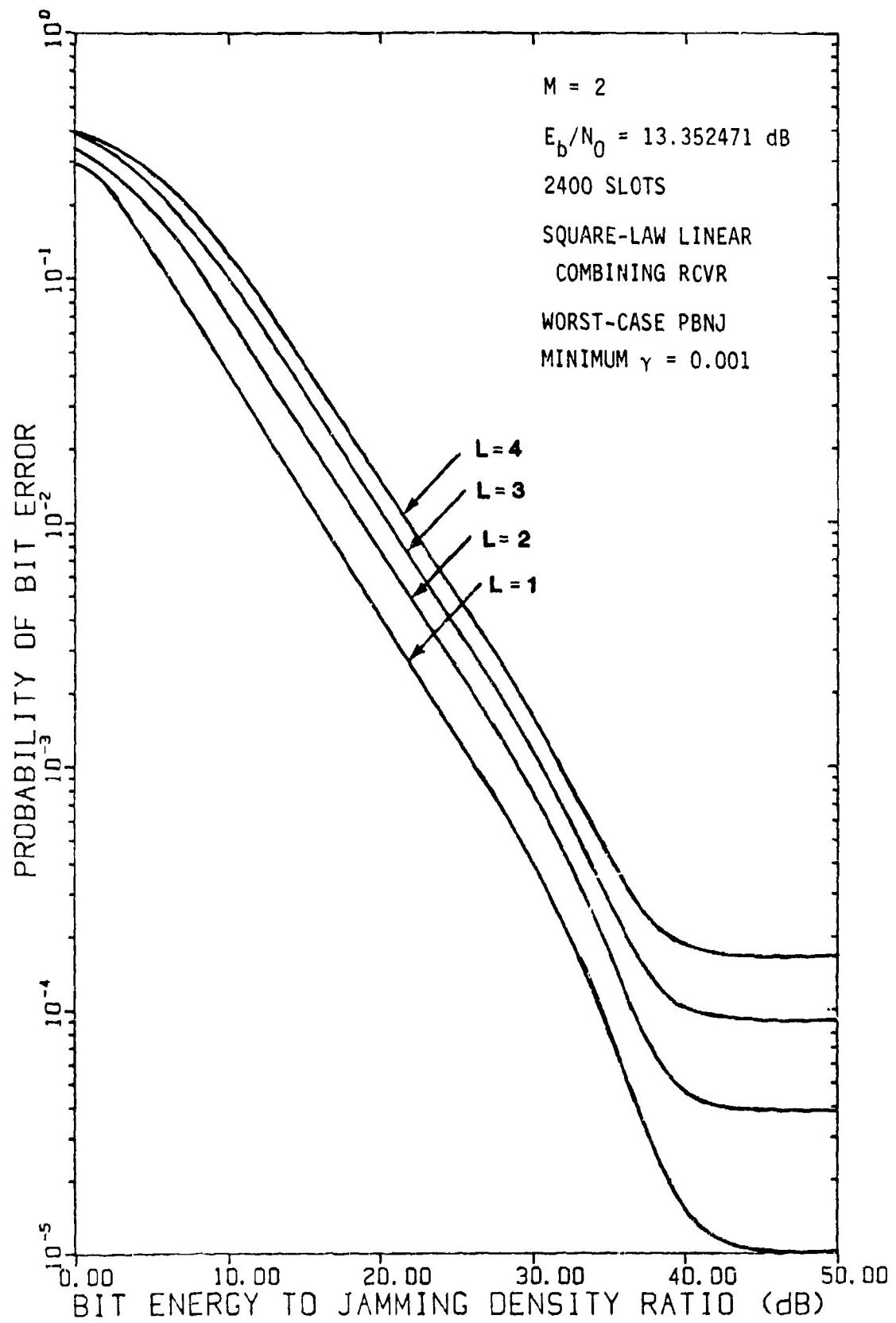


FIGURE 7.4-1 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR SQUARE-LAW COMBINING RECEIVER AND $M = 2$ WITH NUMBER OF HOPS/BIT AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING

J. S. LEE ASSOCIATES, INC.

effect of the jamming after normalization is to reduce the SNR in the MFSK signal channel or slot by an amount which depends upon the random event of that channel's being jammed on ℓ_1 of the L hops. Because the amount of reduction is inversely proportional to γ , the fraction of the system bandwidth which is jammed, while the probability of jamming is directly proportional to γ , there exists an optimum value of γ which maximizes the system error probability as a function of available jamming power; generally $\gamma_{\text{opt}} = \text{const}_1/(E_b/N_j) = \text{const}_2 \cdot J$, that is, the optimum value of γ is directly proportional to jamming power, and for sufficient jammer power full-band jamming ($\gamma=1$) is optimum.

It is possible to reason without analysis that the IC-AGC receiver performs better than the LCR because, while the two receivers are subject to the same SNR degradation, the IC-AGC in effect "matches" the accumulator structure (soft-decision) to the channel. However, it is difficult to predict how any improvement would depend upon L, and whether an optimum value of L exists. Thus the analysis and computations of IC-AGC performance have been quite revealing. For example, Figure 7.4-2 shows that there is a tendency for increasing L to improve the IC-AGC performance for increasing E_b/N_j , but this tendency is by no means uniform for the value of E_b/N_0 shown. Figure 7.4-3 shows the IC-AGC performance which would be obtained if the best value of L were always used. This figure reveals that the effectiveness of the diversity depends on the degree of thermal noise present; when N_0 is not negligible, increasing L eventually gives rise to noncoherent combining losses which overcome the gains from diversity.

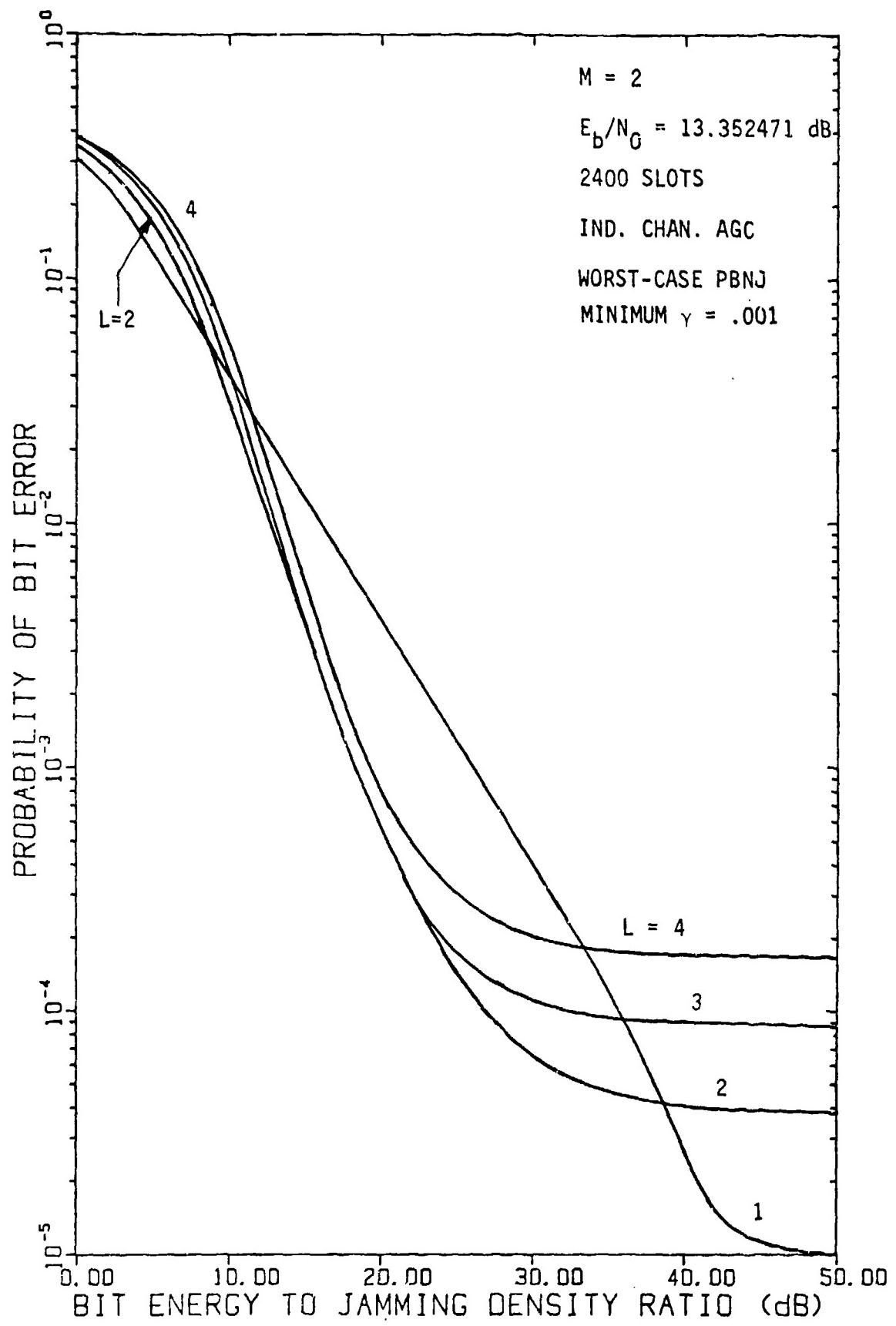


FIGURE 7.4-2 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR INDIVIDUAL CHANNEL AGC RECEIVER AND $M=2$ WITH NUMBER OF HOPS/SYMBOL AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING

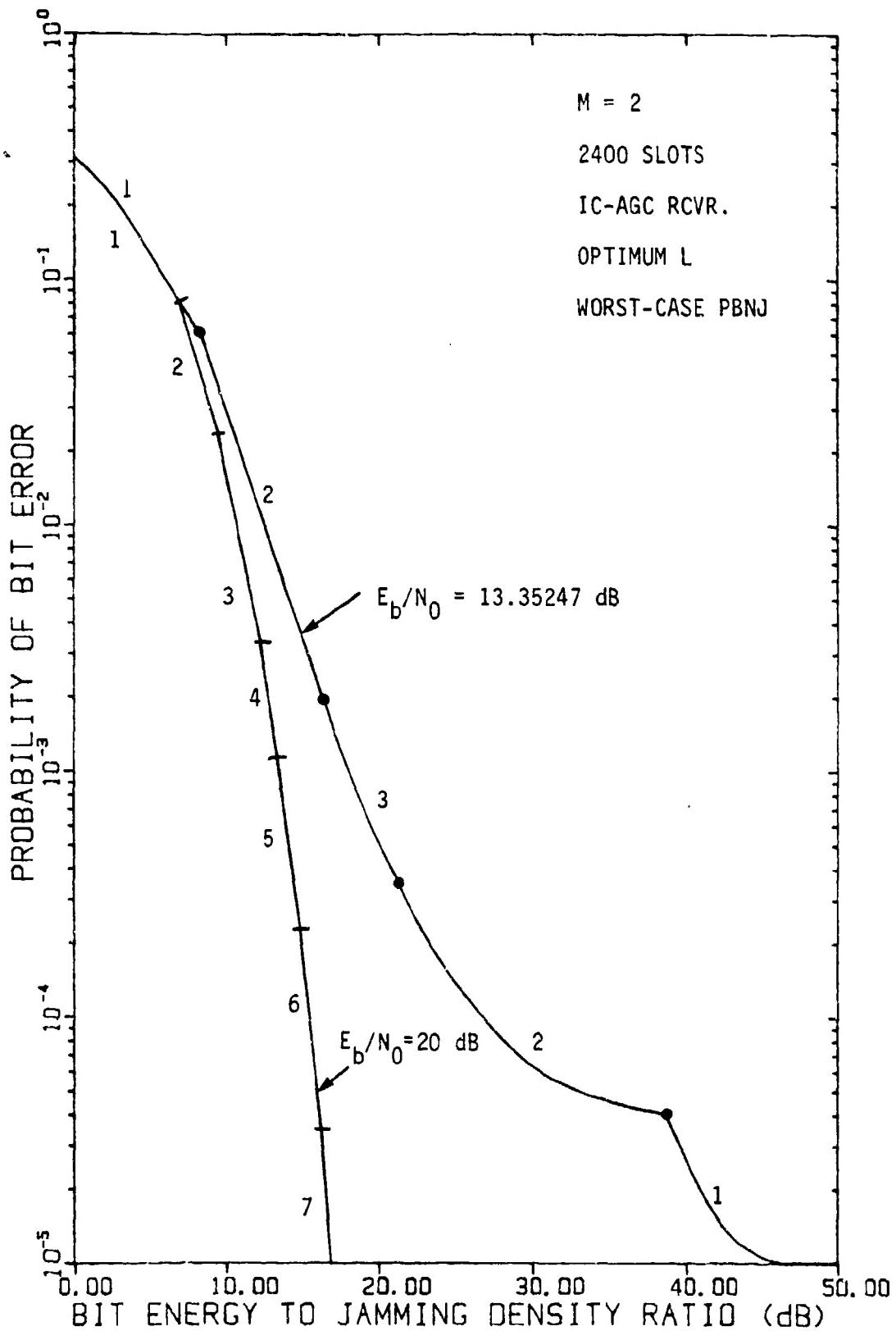


FIGURE 7.4-3 OPTIMUM DIVERSITY PERFORMANCE OF INDIVIDUAL CHANNEL AGC RECEIVER FOR FH/RMFSK WITH $M=2$ AND E_b/N_0 AS A PARAMETER

J. S. LEE ASSOCIATES, INC.

It is important to keep in mind that the performance in jamming can never be better than that without jamming, and that without jamming the best performance is for $L=1$. Therefore "optimum diversity" values may increase with E_b/N_j , but must eventually decrease again to $L=1$ as $E_b/N_j \rightarrow \infty$ (no jamming). However, as the figure demonstrates, for a desired performance of, say 10^{-5} , the optimum diversity value can be greater than one if the unjammed error is much smaller.

Examples of diversity effects for the "any channel jammed" receiver (ACJ) and the self-normalizing receiver are given by Figures 7.4-4 and 7.4-5. We observe that these receivers, in that their normalization techniques "approximate" that of the IC-AGC, achieve similar diversity gain effects.

Two of the receivers studied, the clipper and hard-symbol decision (HD) receivers, do not utilize normalization as such, yet accomplish a diversity gain effect by limiting the "contamination" that a jammed hop may bring to the symbol decision. It has noted previously that the clipper receiver's performance generally is close to that of the ACJ for stronger jamming, but that the HD receiver is considerably worse for the same amount of thermal noise. However, as Figure 7.4-6 shows, even this very simple HD approach can be considered effective in the diversity sense when thermal noise is low.

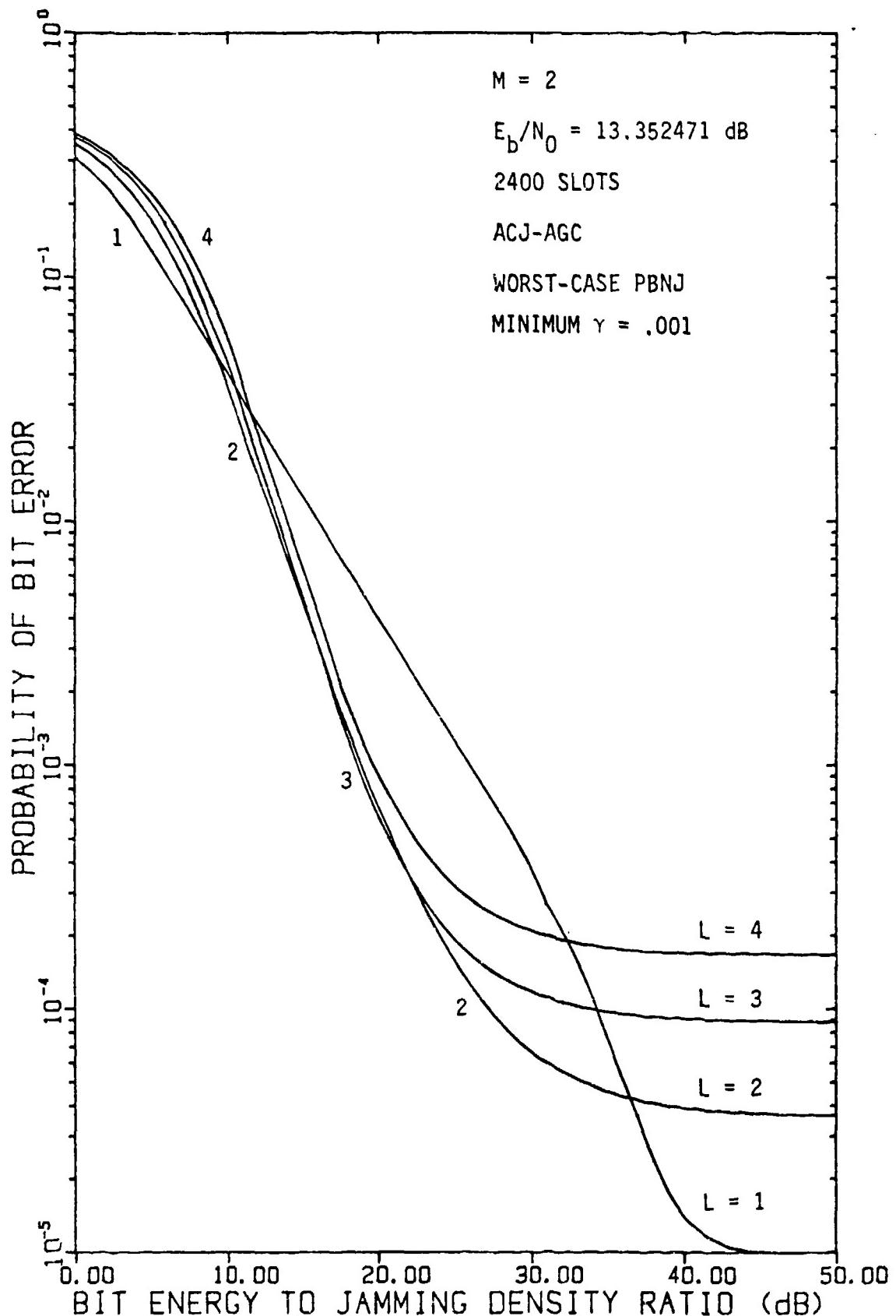


FIGURE 7.4-4 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING DENSITY RATIO FOR ANY-CHANNEL-JAMMED AGC RECEIVER AND $M=2$ WITH NUMBER OF HOPS/BIT AS A PARAMETER IN PRESENCE OF WORST-CASE PARTIAL-BAND NOISE JAMMING

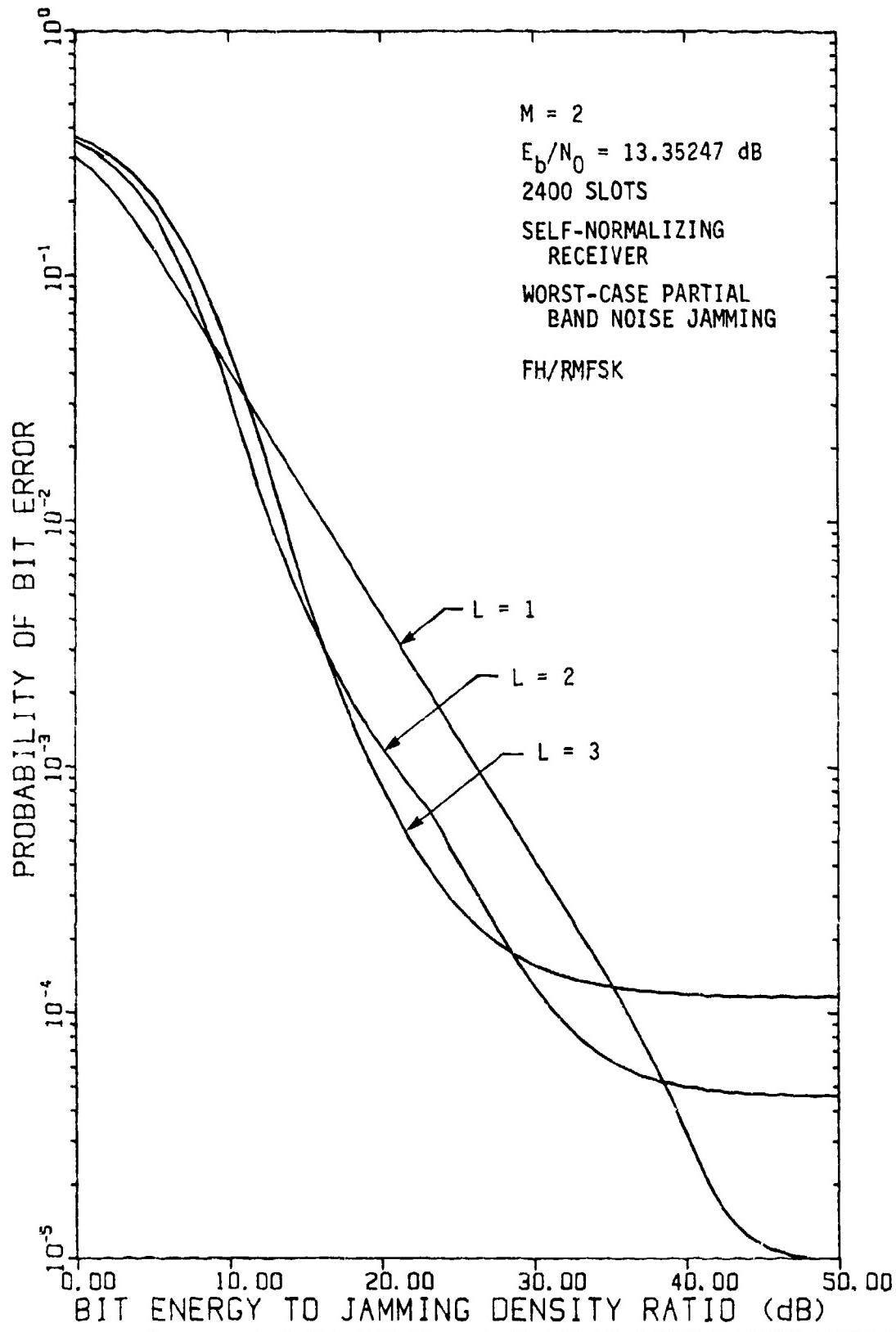


FIGURE 7.4-5 BIT ERROR PROBABILITY VS. BIT ENERGY TO JAMMING NOISE DENSITY RATIO FOR SELF-NORMALIZING RECEIVER IN WORST-CASE PARTIAL-BAND NOISE FOR $M=2$ AND $E_b/N_0=13.35 \text{ dB}$, WITH THE NUMBER OF HOPS/BIT (L) VARIED

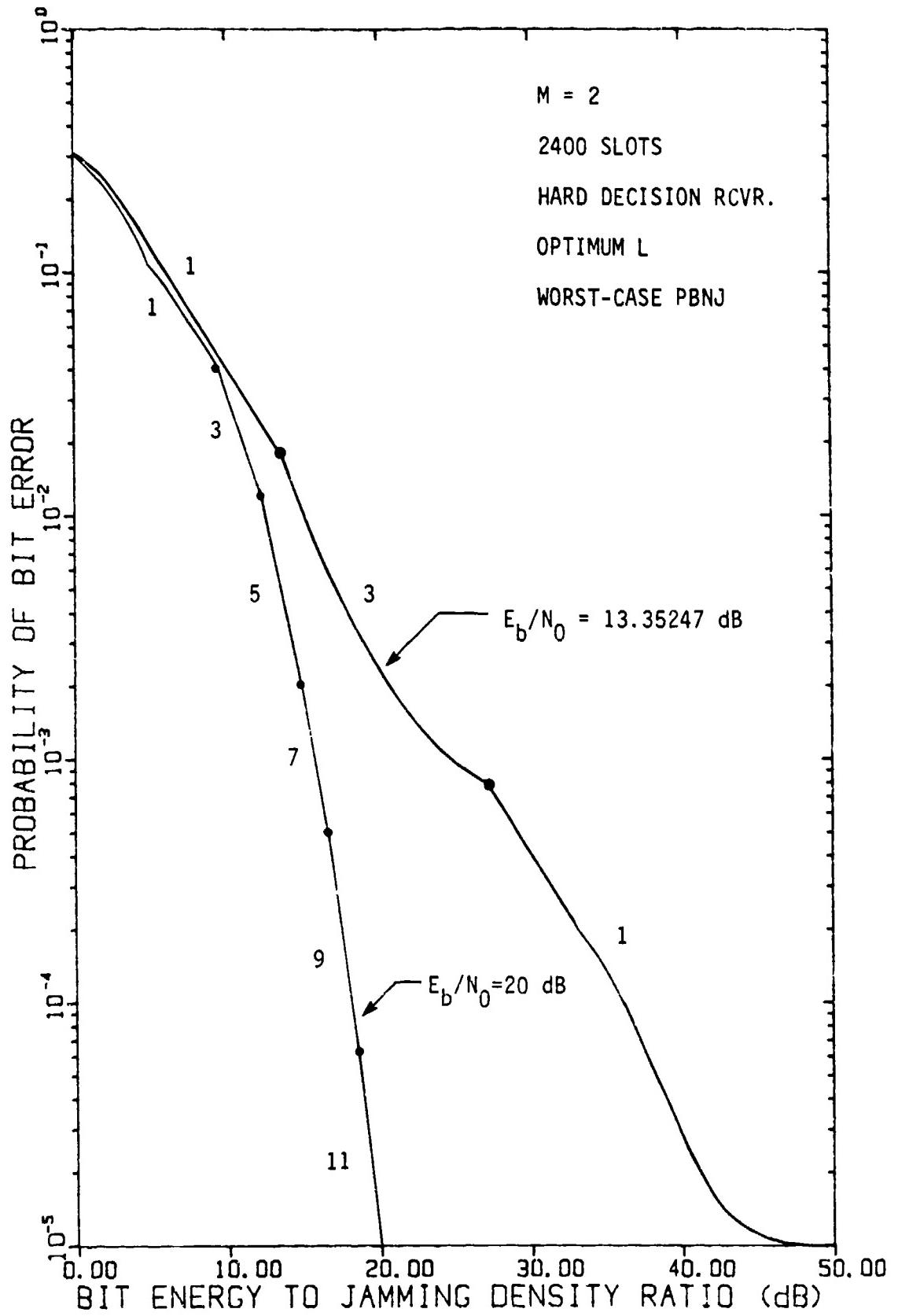


FIGURE 7.4-6 OPTIMUM DIVERSITY PERFORMANCE OF HARD DECISION RECEIVER FOR FH/RMFSK WITH $M=2$

J. S. LEE ASSOCIATES, INC.

8.0 ECCM RECEIVER IMPLEMENTATION STUDIES

In the previous sections, we analyzed the BER performance of various ECCM receiver processing schemes for uncoded FH/RMFSK radio systems in the presence of worst-case partial-band noise jamming (PBNJ).

Our objective has been to provide a comparison of these different systems, which vary extensively in their implementation complexities. These results will enable the ECCM system designer to weigh the engineering cost requirements of a particular receiver design versus the anti-jam effectiveness. Toward this end, we now explore practical issues related to the implementation of these different processing schemes along with an assessment of implementation effects.

8.1 IMPLEMENTATION ISSUES AND CONCEPTS

All receivers suppress the total noise jamming power by an amount equivalent to the system processing gain, defined as the ratio of the jammer bandwidth to the receiver bandwidth. Hopping the signal forces the jammer to spread its power over a wide bandwidth, but the jammer can maximize its effectiveness by selecting an optimum bandwidth, which is a certain fraction (γ) of the total hopping system bandwidth (W). This results in a BER which tends to be an inverse linear function of E_b/N_j , so that more than 40 dB of E_b/N_j is required to obtain BER's less than 10^{-4} . ECCM FH/MFSK and FH/MFSK systems counter this effect by using multiple hops per symbol, with the L hops per symbol combined at the receiver as in diversity transmission schemes.

We have demonstrated that effective jammer suppression is obtained by incorporating a nonlinear function in each M-ary channel prior to combining. The improvement in BER performance is realized by the fact that within a PBNJ environment, the nonlinear techniques (clipper, AGC, hard-decision, SNORM) mitigate the tendency of a jammed hop to dominate the symbol decision. Of these nonlinear

J. S. LEE ASSOCIATES, INC.

techniques we have studied, it was assumed that certain a priori information or perfect measurements are available to the clipper (SNR threshold) and AGC (noise powers) schemes; no such measuring tactics are necessary for the hard-decision or SNORM receivers. In what follows, we investigate the practicalities concerning the implementation and impact of non-ideal noise power measurements and threshold settings.

8.1.1 ECCM Receiver Information Requirements

In Table 8.1-1 we summarize the ECCM techniques used by the various receivers we have studied, including the information necessary for their implementation. The square-law linear combining receiver is presented as a baseline for comparison with the other, nonlinear combining types. Our specific interest here is to address the feasibility of implementation.

In the table, the nonlinear combining receivers are classified according to whether their anti-jam measures operate on a per-symbol basis (across all M channels) or on a per-channel basis. The per-symbol ECCM receivers include the ACJ-AGC, the hard-decision, and the SNORM receivers. Of these, the ACJ-AGC is seen as the only type utilizing a priori information on the received noise (thermal plus jamming), since it weights all channels on a given hop by

$$w_{mk} = w_k = \begin{cases} (\sigma_N^2)^{-1}, & \text{no channels jammed on hop } k \\ (\sigma_T^2)^{-2}, & \text{any channel jammed on hop } k \end{cases}$$
$$= \left[\max_m (\sigma_{mk}^2) \right]^{-1}. \quad (8.1-1)$$

TABLE 8.1-1 SUMMARY OF ECCM TECHNIQUES FOR THE RECEIVERS STUDIED

EXTENT OF HOP WEIGHTING OR ANTI-JAM-MEASURE	RECEIVER(S)	SUPPRESSION TECHNIQUE	INFORMATION REQUIRED
None	Linear Combining	None	None
Per-Symbol	Any-Channel 1-Jammed AGC (ACJ-AGC)	Scale Down Jammed Hops (Normalize Max Variance to 1)	$\max \sigma_{mk}^2$
	Hard-Symbol Decision	Limit Hops to One Vote	None
	Self-Normalizing (SNORM)	Scale Down Jammed Hops (Normalize Channel Sum to 1)	None
Per-Channel	Individual Channel AGC (IC-AGC)	Scale Down Jammed Channels (Normalize All Variances to 1)	σ_N^2
	Clipper	Limit Each Channel to Max Value of n	σ_N^2 and E_b/N_0 (to Set Value of n)

*Notes: (1) Hops numbered by k ($k=1, 2, \dots, L$) and channel's numbered by m ($m=1, 2, \dots, M$)

(2) σ_{mk}^2 is the a priori variance of the total noise present in channel m at time k

(3) Since σ_{mk}^2 is assumed to be either σ_N^2 (unjammed) or σ_f^2 (jammed), "information required" may be interpreted as knowing whether a channel is jammed (IC-AGC) or a symbol is jammed (ACJ-AGC)

J. S. LEE ASSOCIATES, INC.

Therefore, the information required is $w_k^{-1} = \max_m (\sigma_{mk}^2)$, which involves knowing σ_N^2 , σ_T^2 , and whether any of the M channels is jammed. As a minimum, the ACJ receiver needs to know the ratio $\max_m (\sigma_{mk}^2) / \sigma_N^2$, since the operation of the receiver is unaffected if unjammed hops are left alone (weight = 1) and jammed hops are reduced by the factor σ_N^2 / σ_T^2 .

The hard-decision receiver is classified as a per-symbol ECCM receiver because, as we have observed in previous sections, its operation in effect limits each symbol piece (hop) to one vote in the M-ary majority logic decision, no matter how strongly a hop may have been jammed. Its operation does not require any a priori information or measurement.

The SNORM receiver derives its per-symbol weights from the M square-law envelope detector samples themselves:

$$w_k = \left(\sum_m x_{mk}^2 \right)^{-1}. \quad (8.1-2)$$

Therefore, it does not require a priori information or additional measurements in its operation.

The per-channel ECCM receivers include the IC-AGC and the clipper receivers, and both utilize a priori information. The IC-AGC weights each channel sample by the inverse of its a priori noise variance:

$$w_{mk} = (\sigma_{mk}^2)^{-1}. \quad (8.1-3)$$

In this manner, all channels on all hops are normalized to have unit noise variance; any channels which are jammed ($\sigma_{mk}^2 = \sigma_T^2$) are therefore suppressed. This technique involves knowing σ_N^2 , σ_T^2 , and the jamming state or condition

of each channel. Alternately, the ratios σ_{mk}^2/σ_N^2 are needed, as a minimum.

The clipper receiver achieves an ECCM effect by "containing" any jammed channels; their contribution to the soft-decision sums cannot be any larger than the clipping threshold (n), no matter how strongly jammed. In order to set the threshold, both σ_N^2 and E_b/N_0 a priori values are needed since $n = n(\sigma_N^2, E_b/N_0)$ is chosen to minimize the error without jamming.

With respect to the additional receiver complexity required to develop information needed by the suppression technique, only the AGC schemes (jamming decision and normalization weights) and clipper (SNR levels) receivers need be addressed.

8.1.2 Measurement Approaches.

Implementation of the two AGC schemes requires differentiation between two zero-mean bandpass Gaussian noise processes with different variances which determine a jammed/unjammed channel state. In addition, to be useful as a quantity for a normalization weight in an AGC scheme, our measurement (noise-power estimate) must reflect as closely as possible in real-time the actual system state of the measured channel. This leads us to consider factors in both time and frequency domain representations of our measuring technique, i.e. the accuracy or quality of a band-limited channel noise-power measurement.

It is assumed throughout the measurement process that the data measures are sample records from a continuous stationary random process. Letting $x(t)$ be a single sample time history record from a zero-mean stationary (ergodic) Gaussian random process $\{x(t)\}$, the mean-square value (variance) of $\{x(t)\}$ can be estimated by time-averaging over a finite time interval τ as

$$\hat{\sigma}_x^2 = \frac{1}{\tau} \int_0^\tau x^2(t)dt \quad (8.1-4)$$

J. S. LEE ASSOCIATES, INC.

with the true mean-square value being

$$\sigma_x^2 = E [x^2(t)] \quad (8.1-5)$$

and is independent of t since $\{x(t)\}$ is stationary. Now the expected value of the estimate of σ_x^2 is

$$E[\hat{\sigma}_x^2] = \frac{1}{\tau} \int_0^\tau E[x^2(t)] dt = \frac{1}{\tau} \int_0^\tau \sigma_x^2 dt = \sigma_x^2. \quad (8.1-6)$$

Thus, $\hat{\sigma}_x^2$ is an unbiased estimate of σ_x^2 , independent of τ .

Regarding the quality of measurement, it is shown [19, p. 176] that the variance of a mean-square value estimate of band-limited white Gaussian noise with zero mean is

$$\text{Var}[\hat{\sigma}_x^2] \approx \frac{\sigma_x^2}{B\tau}. \quad (8.1-7)$$

Hence, we clearly see the inverse relationship of accuracy of our measurement to the measured channel time-bandwidth product; that is, for $B\tau \rightarrow \infty$ we would realize a "perfect" noise measurement.

8.1.2.1 A look-ahead measurement scheme.

Since the optimum power estimator uses a square-law detector [18] one could obtain a workable noise-power estimate by measuring the next slot to be hopped into by all M-ary channels; such a "look-ahead" scheme is illustrated in Figures 8.1-1 and 8.1-2.

Figure 8.1-1 shows the first part of the scheme in which we obtain the look-ahead receiver samples. Here we allow the receiver channel code-synchronized PN sequence generators to be delayed by one hop period (τ) in relationship to the respective measurement channel synchronization. In this manner we obtain the look-ahead samples at hop time k for use with

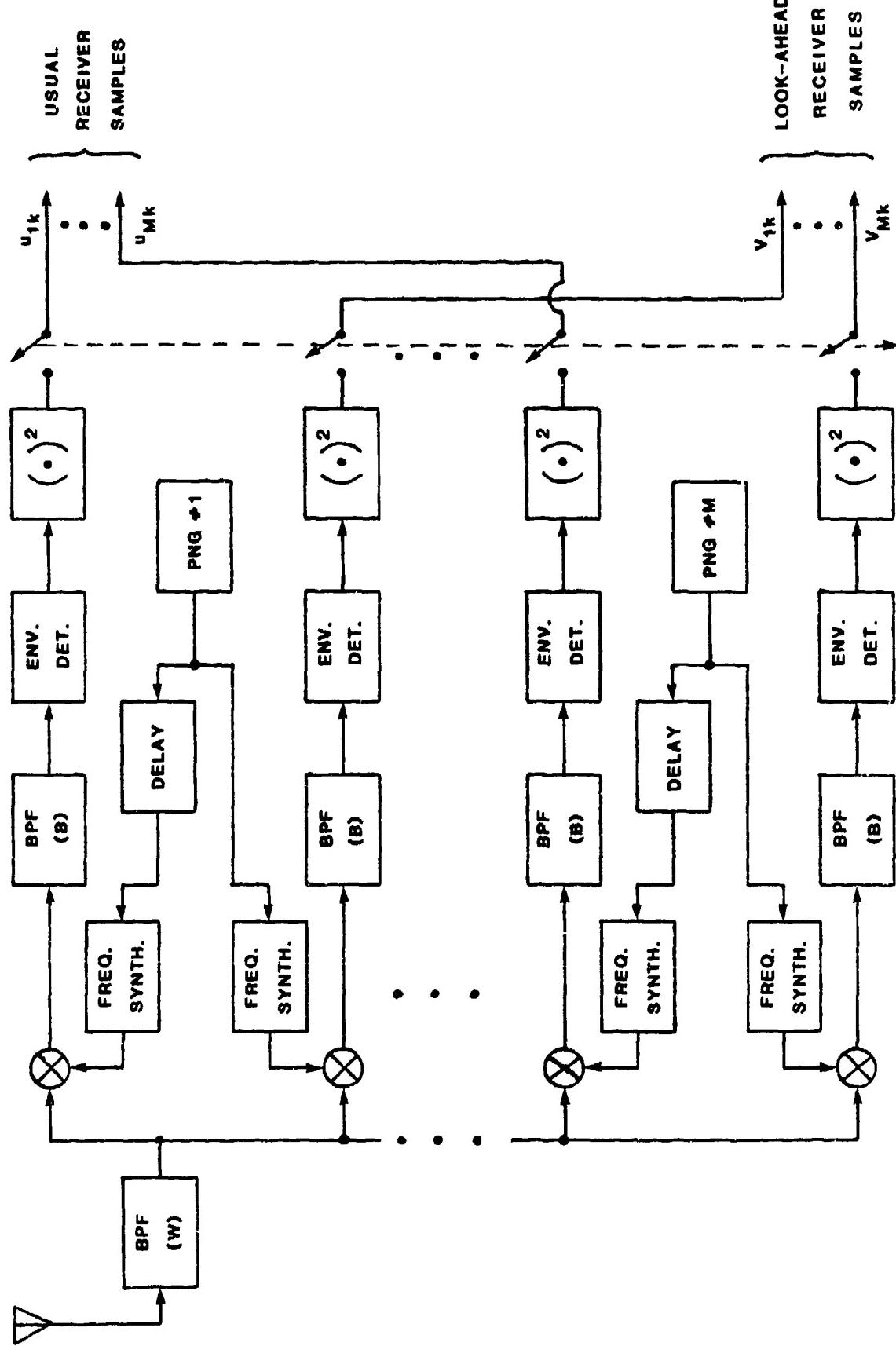


FIGURE 8.1-1 ACQUISITION OF JAMMING STATE DATA BY A "LOOK-AHEAD" SCHEME

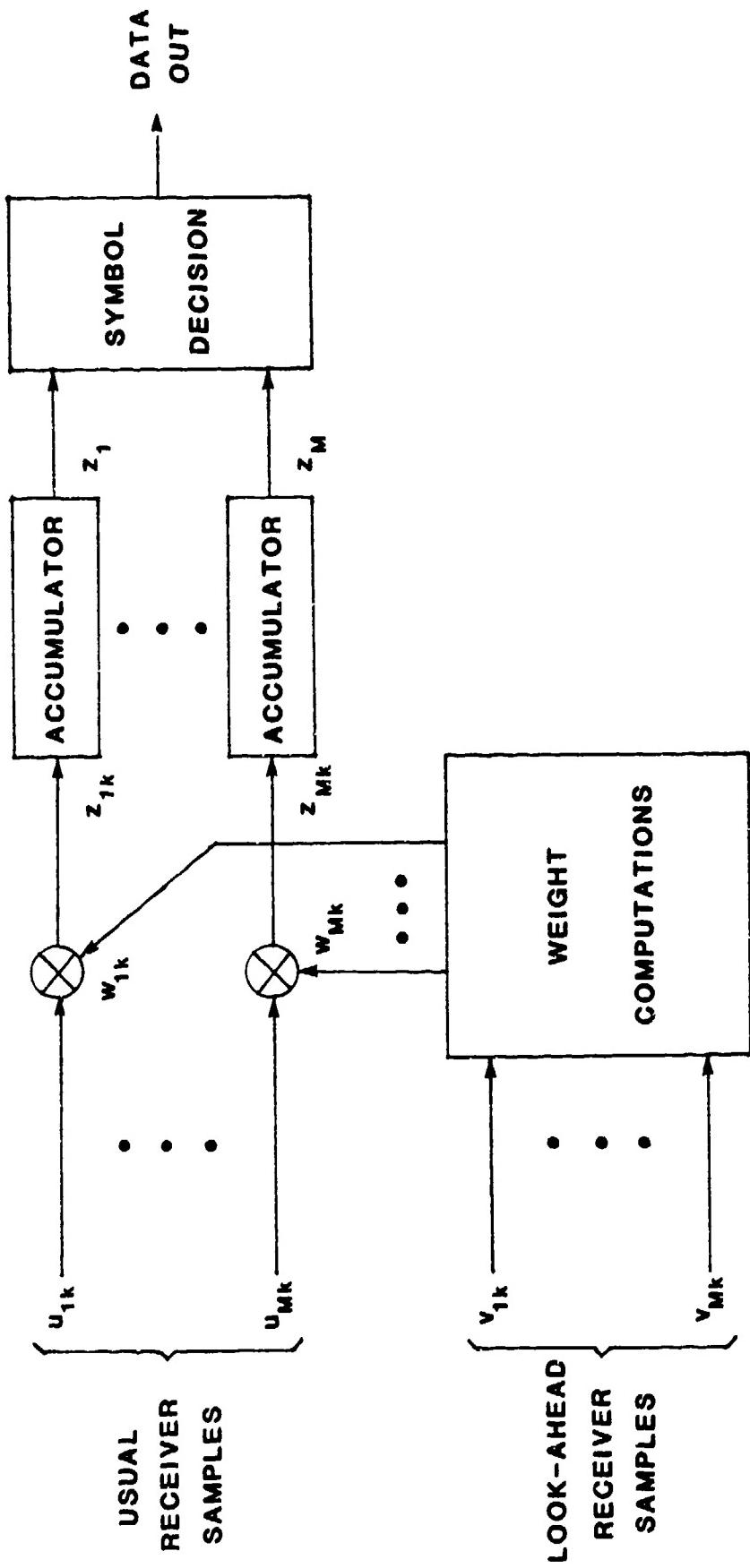


FIGURE 8.1-2 USE OF LOOK-AHEAD MEASUREMENTS TO DERIVE ECCM WEIGHTS

J. S. LEE ASSOCIATES, INC.

the usual communicator receiver samples to be gathered at time $k+1$. We note that the look-ahead samples are assumed to be values of either σ_N^2 or σ_T^2 and that any type of spectral interference from other hoppers in the network is nonexistent. Also, this first part (Figure 8.1-1) of the look-ahead operation is the same for both IC- and ACJ-AGC receiver types.

In Figure 8.1-2 we show the use of the M-channel look-ahead measurements to derive the proper normalization weights. Thus, at time $k+1$ we obtain the following variables for accumulating in the L-hop diversity combining state for each of the M channels:

$$z_{mk} = x_{mk}^2 \cdot [\max_m (\hat{g}_{mk}^2)]^{-1}, \text{ for ACJ} \quad (8.1-8)$$

$$z_{mk} = x_{mk}^2 \cdot (\hat{\sigma}_{mk}^2)^{-1}, \text{ for IC.} \quad (8.1-9)$$

Additionally, the weight computation network could incorporate a multi-hop/multi-channel processing stage (see Figure 8.1-3) to determine jamming state information (JSI) based upon multiple look-ahead measurements. Should the channel be jammed, we then have a one-sample estimate of the noise plus jamming power $\hat{\sigma}_T^2$. If unjammed, we obtain $\hat{\sigma}_N^2$ which is a one-sample measure of the channel's thermal noise power. The additional processing envisions each channel estimate $\hat{\sigma}_N^2$ as going into a smoothing operation, which would then output the improved σ_N^2 and σ_T^2 estimates. These values in turn are fed back to the jammed/unjammed decision blocks in the form of the detection threshold. The use of multiple-hop/multiple-channel measurements improves the quality of the noise power estimates by increasing the $B\tau$ product in (8.1-7). Details involved in both the jamming state detection and data smoothing of estimates are discussed later in this section.

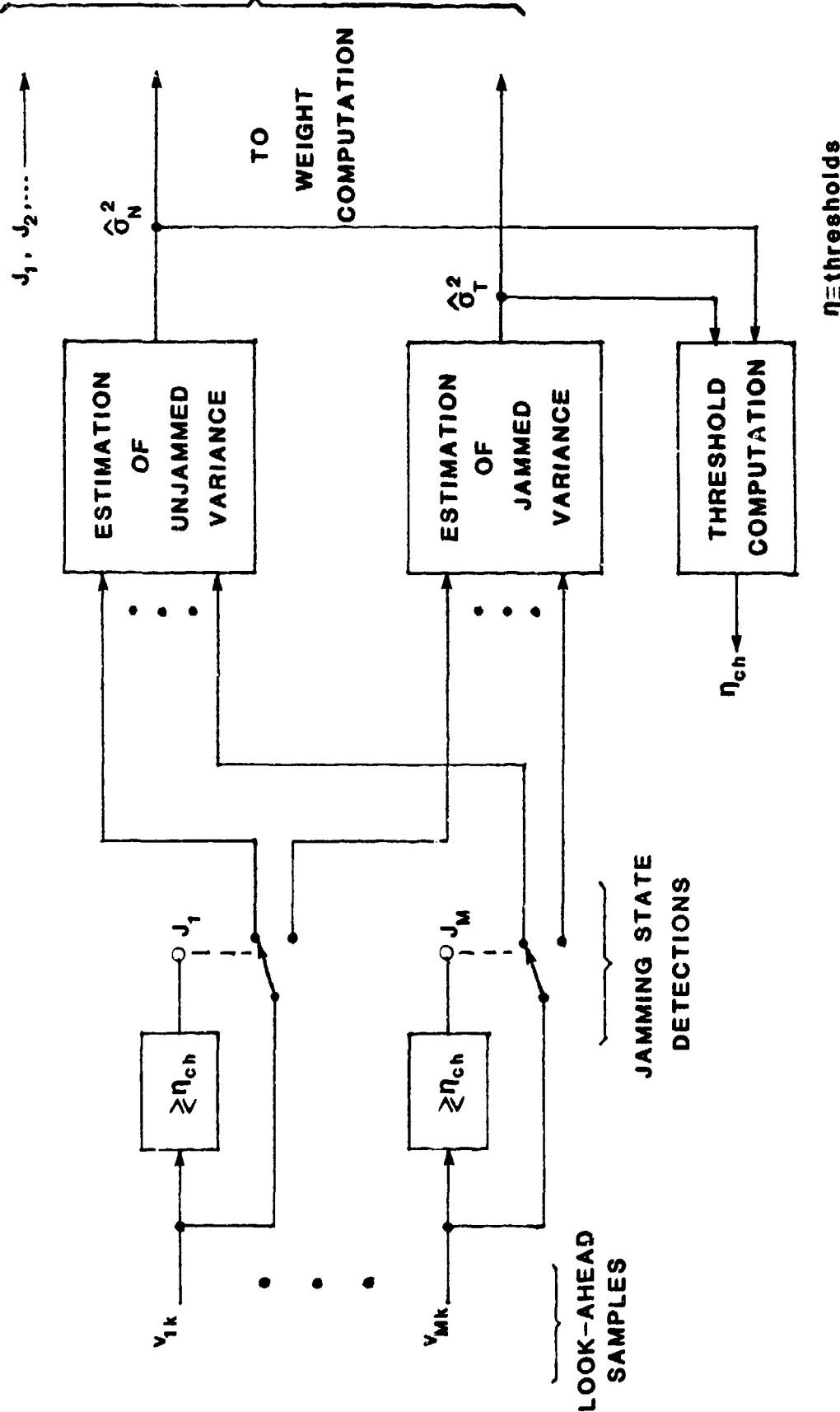


FIGURE 8.1-3 POSSIBLE USE OF MULTIPLE-HOP/MULTIPLE CHANNEL SAMPLES TO IMPROVE MEASUREMENTS

J. S. LEE ASSOCIATES, INC.

8.1.2.2 In-band measurement schemes.

One in-band measurement scheme assumes that $\hat{\sigma}_N^2$ and $\hat{\sigma}_T^2$ are to be measured between signal transmission times over the frequency-hopping bandwidth W . That is, a measurement process takes place during the communications "link idle" state, enabling the gathering of measurement samples over many hop periods. This procedure avoids the system complexity required by the look-ahead scheme, and is subject to the same caveats regarding stationarity of the noise (thermal, background, and jamming) environment across the hopping system bandwidth, as well as the corruption of purely noise estimates by the activity of other communications users during measurement intervals.

Receiver frame synchronization information or message preamble data could be used to decide when to cease or start taking measurement data samples. In concept, these measurements would enable high quality estimates of σ_N^2 and σ_T^2 to be developed, and/or perhaps even a stored estimate of the received noise power as a function of frequency.

Now, in order to detect the presence of jamming or to set the clipper's optimized threshold, we require knowledge of the received (average) thermal noise power $\sigma_N^2 = N_0 B$, where N_0 is the noise density in watts per hertz, assumed to be independent of frequency, and B is the channel bandwidth. In concept, an independent channel noise-power measurement system could be used to take several simultaneous measurements uniformly distributed within the thermal-noise-only (unjammed) portion of the hopping band W . The arithmetic average of these measurements would then be the estimate $\hat{\sigma}_N^2$. Assuming that σ_N^2 varies slowly, if at all, we would continuously correct the long-term moving average of $\hat{\sigma}_N^2$;

J. S. LEE ASSOCIATES, INC.

that is, a type of smoothing operation in which the processing scheme uses all measurements between times 0 and T to estimate the state of a system at a time t, where $0 < t < T$. This smoothed estimate of σ_N^2 over the time interval 0 and T can be denoted by $\sigma_N^2(t|T)$. Specifically, we envision a fixed-lag smoother in which a running smoothing solution lags the most recent measurement by a constant time delay Δ and is denoted as $\hat{\sigma}_N^2(T-\Delta|T)$. A reasonable value of Δ would be equal to the time for one symbol transmission.

Figure 8.1-4 illustrates a conceptual in-band measurement approach for the AGC type FH/RMFSK receivers. There are two modes: (a) between signal transmissions and (b) during signal transmissions. Between transmissions, the receiver continues to operate its synthesizers, detectors, and samplers to gather data for estimates of σ_N^2 and σ_T^2 , as mentioned above. During transmissions, jamming detection at the channel level (threshold n_{ch}) or symbol level (threshold n_{sym}) would furnish jamming state information (JSI) for selection of AGC weights, using thresholds based on the estimates of σ_N^2 and σ_T^2 . Possibly the samples received during transmissions could be used also for the variance estimation by feeding back the symbol decision to identify the channel with the signal, as suggested in the diagram.

With respect to the clipper receiver, Figure 8.1-5 depicts a scheme for setting the optimized clipping threshold n_0 . Toward this end, we need to obtain a current estimate of the clipper receiver's SNR. Similar to the previously described two-mode in-band noise-power measurement concept for the AGC receivers, an individual channel measurement system would likewise be used to estimate the received signal power S. Several measurements would be taken

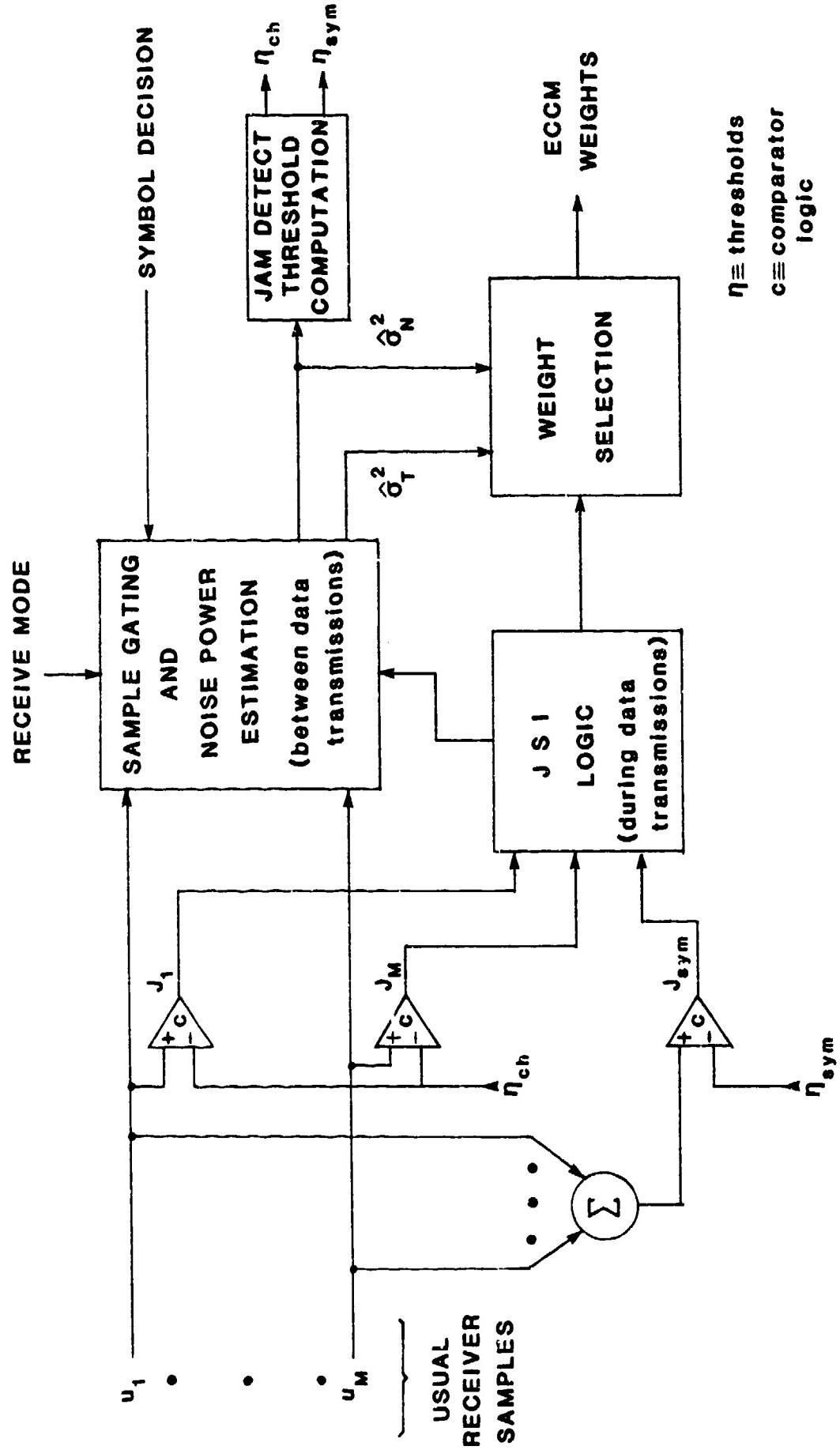


FIGURE 8.1-4 CONCEPTUAL PROCESSING OF IN-BAND MEASUREMENTS
TO IMPLEMENT AGC RECEIVERS

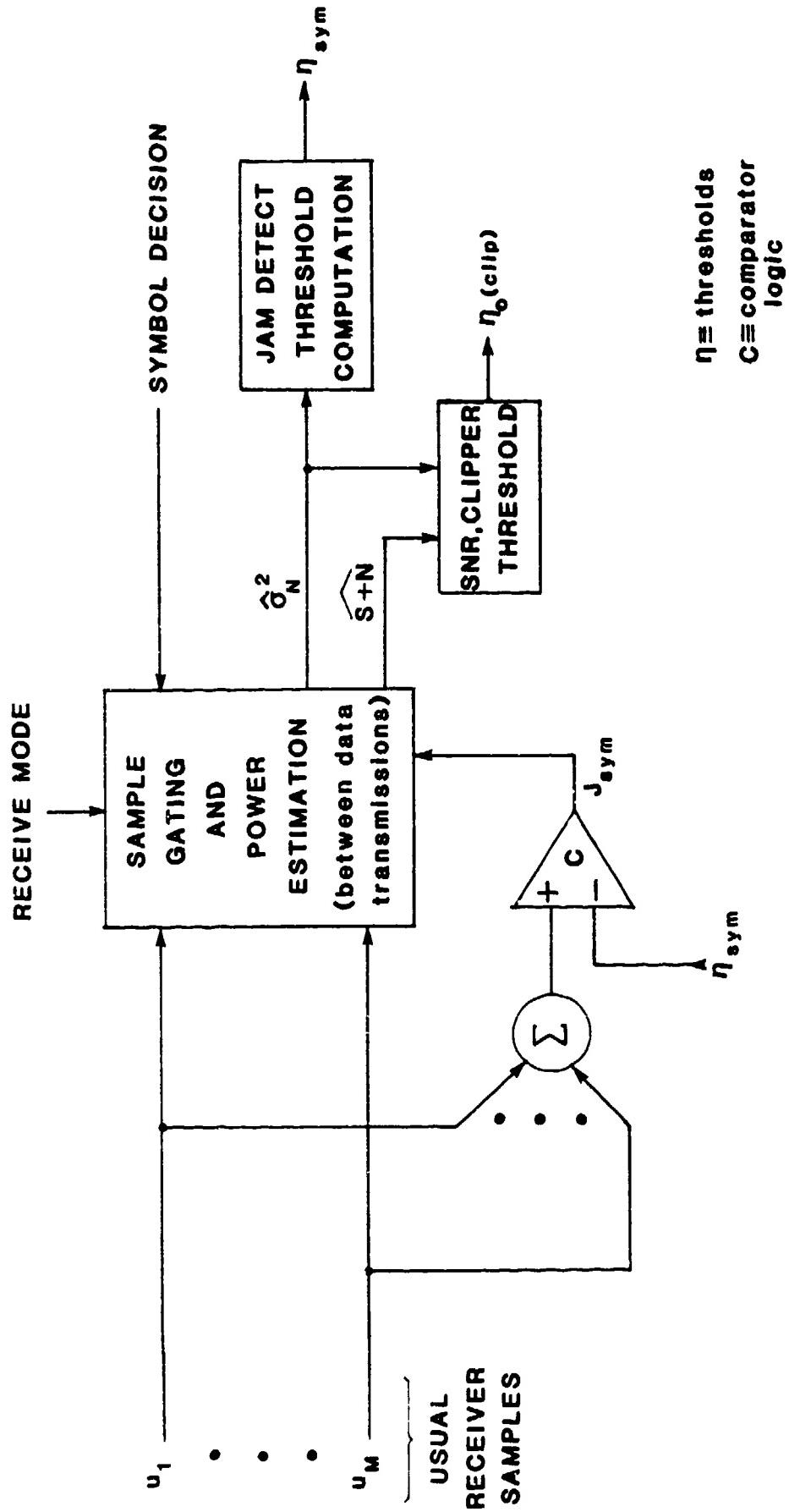


FIGURE 8.1-5 PROCESSING OF IN-BAND MEASUREMENTS
TO IMPLEMENT CLIPPER RECEIVER

of the received signal plus thermal noise within the unjammed portion of the hopping band W , using symbol decision feedback to identify the signal channel. The arithmetic average of these measurements forms an estimate of signal plus noise power, \hat{P}_{sn} . This estimate could also be refined by a fixed-lag (per-symbol) smoothing operation. Hence, we would obtain

$$\hat{\rho}_N = \frac{\hat{S}}{\hat{\sigma}_N^2} = \frac{\hat{P}_{sn}}{\hat{\sigma}_N^2} - 1 = \frac{K}{L} \cdot \frac{\widehat{E_b}}{N_0}, \quad (8.1-10)$$

the signal-to-noise ratio for a given hop dwell time, to be updated on a per-symbol basis.

Table 5.3-1 showed a summary of the clipper receiver values of n_0 for a given L , M , and E_b/N_0 . Note that these numeric values were obtained only for values of E_b/N_0 such that $P_b(e) = 10^{-5}$ for two MFSK systems in the absence of jamming. We point out that, in practice, new values of n_0 need to be computed for each different value of E_b/N_0 . Therefore the actual clipper receiver would require that matrices of n_0 values be stored in a read-only memory (ROM) look-up table.

8.1.3 Jamming State Decisions.

Implementation of a jamming state decision scheme would be based upon the following assumptions: (1) that a look-ahead or in-band measurement scheme is utilized, and (2) that the noise power estimates of $\hat{\sigma}_N^2$ and $\hat{\sigma}_T^2$ are readily available in time. Ultimately, the criterion for making a jammed/unjammed state detection is predicated upon a single observation (per channel) or a multiple observation (per symbol). The ACJ-AGC receiver requires detection of a jammed symbol hop whereas the IC-AGC requires detection of jamming in each of the M channels on each hop.

J. S. LEE ASSOCIATES, INC.

In both the symbol and the channel cases, the problem is one of deciding between Gaussian noise with variance σ_N^2 and Gaussian noise with variance σ_T^2 . Due to the lack of a priori jamming probabilities (γ) or cost functions, we utilize the Neyman-Pearson criterion as our hypothesis testing technique. Its test objective is to maximize the probability of detection (P_D) for a given probability of false alarm (α) and is accomplished by employing a likelihood ratio test.

8.1.3.1 Jammed channel detection.

A basic problem in determining a jammed/unjammed state with an individual channel look-ahead scheme is in accounting for the signal itself. Recalling that the look-ahead measurement channel is one hop period "ahead" of the normal receiver channel, the situation can arise in which the measurement channel is actually the present signal channel. However, we first analyze the case for a measured channel without a signal present.

Using the narrowband Gaussian process representation for the channel samples, the hypotheses to be considered are

$$H_0: p(\underline{n}_c, \underline{n}_s) = \frac{1}{2\pi\sigma_N^2} \exp \left\{ -\frac{\underline{n}_c^2 + \underline{n}_s^2}{2\sigma_N^2} \right\} \quad (8.1-11)$$

$$H_1: p(\underline{n}_c, \underline{n}_s) = \frac{1}{2\pi\sigma_T^2} \exp \left\{ -\frac{\underline{n}_c^2 + \underline{n}_s^2}{2\sigma_T^2} \right\} \quad (8.1-12)$$

J. S. LEE ASSOCIATES, INC.

where H_0 is the unjammed case and H_1 is the jammed case. For the likelihood ratio we obtain

$$\Lambda = \frac{\sigma_N^2}{\sigma_T^2} \exp \left\{ - \frac{(n_c^2 + n_s^2)}{2} \left(\frac{1}{\sigma_T^2} - \frac{1}{\sigma_N^2} \right) \right\} \quad (8.1-13)$$

and in comparing the log-likelihood function to a threshold we have

$$\log \Lambda = \log \left(\frac{\sigma_N^2}{\sigma_T^2} \right) + \frac{x^2}{2} \left(\frac{\sigma_T^2 - \sigma_N^2}{\sigma_N^2 \sigma_T^2} \right) \stackrel{H_1}{\geq} \stackrel{H_0}{n} \quad (8.1-14)$$

where $x^2 = n_c^2 + n_s^2$ is our estimated look-ahead noise-power measurement or sample test statistic for a single channel.

The value of x^2 to decide that jamming is present (H_1 true) is

$$x^2 \geq \frac{2\sigma_N^2 \sigma_T^2}{\sigma_T^2 - \sigma_N^2} \left[n - \log \left(\frac{\sigma_T^2}{\sigma_N^2} \right) \right] = n_{ch}, \sigma_T^2 > \sigma_N^2. \quad (8.1-15)$$

For the false alarm probability we have

$$P_{FA} = \Pr \left\{ x^2 > n_{ch} | H_0 \right\} = \alpha = e^{-n_{ch}/2\sigma_T^2}. \quad (8.1-16)$$

Similarly, the probability of detection is

$$P_D = e^{-n_{ch}/2\sigma_T^2} \quad (8.1-17)$$

with a receiver operating characteristic for our jamming detector given by

$$P_D = \left[\alpha \right]^{\sigma_N^2 / \sigma_T^2} = \left[\alpha \right]^{1/K}. \quad (8.1-18)$$

J. S. LEE ASSOCIATES, INC.

Now in the case of the look-ahead channel sampling a signal channel slot, these samples (x^2) are independent noncentral chi-squared random variables with two degrees of freedom, multiplied by the total channel noise power σ_{ch}^2 , and with noncentrality parameters

$$\lambda = \frac{2S}{\sigma_{ch}^2} = \begin{cases} 2S/\sigma_N^2 = 2\rho_N, & \text{hop unjammed} \\ 2S/\sigma_T^2 = 2\rho_T, & \text{hop jammed.} \end{cases} \quad (8.1-19)$$

Therefore, the pdf of a given sample $u=x^2$ is

$$f_{sig}(u) = \frac{1}{2\sigma_{ch}^2} e^{-(u+2S)/2\sigma_{ch}^2} I_0(\sqrt{2Su}/\sigma_{ch}^2) \quad (8.1-20a)$$

$$= \begin{cases} (1/2\sigma_N^2) e^{-(u+2S)/2\sigma_N^2} I_0(\sqrt{2Su}/\sigma_N^2), & u \geq 0, \text{ hop unjammed} \\ (1/2\sigma_T^2) e^{-(u+2S)/2\sigma_T^2} I_0(\sqrt{2Su}/\sigma_T^2), & u \geq 0, \text{ hop jammed} \end{cases} \quad (8.1-20b)$$

where S signifies power in the signal itself. Consequently, the probability of a false alarm and the probability of detection can be written as

$$P_{FA} = Q(\sqrt{2\rho_N}, \sqrt{n_{ch}/\sigma_N^2}) \quad (8.1-21)$$

$$P_D = Q(\sqrt{2\rho_T}, \sqrt{n_{ch}/\sigma_T^2}) \quad (8.1-22)$$

where $Q(x,y)$ is Marcum's Q-function.

J. S. LEE ASSOCIATES, INC.

8.1.3.2 Jammed symbol detection.

In the jammed M-ary symbol detection case, we have the two hypotheses of: (1) no channel is jammed, and (2) any of the M channels is jammed, i.e. ACJ. Hence, the situation is one of multiple alternative hypothesis testing given as

$$H_0: \sigma_m^2 = \sigma_N^2, \quad m = 1, 2, \dots, M; \quad (8.1-23)$$

$$H_1: \sigma_m^2 = \sigma_T^2, \quad 2^M - 1 \text{ possible combinations of jammed channels};$$

where σ_m^2 is a parameter in the likelihood function of the measurement samples which is written as

$$p(n_c, n_s) = \prod_m \left(\frac{1}{2\pi\sigma_m^2} \right) e^{-x_m^2/2\sigma_m^2}. \quad (8.1-24)$$

As in the jammed channel detection case, we require a current estimate of σ_N^2 obtained from a look-ahead or in-band noise-power measurement scheme. Furthermore, an estimate of SNR ($\hat{\rho}_N$) is also needed for threshold determination when accounting for the signal itself being detected in a look-ahead scheme.

Since there are many alternatives to H_0 as expressed by (8.1-23), we consider the simplified test of whether the sum of the channel samples exceeds a threshold.

For the case of the signal being present in one of the channels, the sum of the samples ($u_m = x_m^2$) is distributed as, with no jamming,

$$\sum_{m=1}^M u_m = \sigma_N^2 \chi^2(2M, 2\rho_N) \quad (8.1-25)$$

J. S. LEE ASSOCIATES, INC.

i.e., a non-central chi-squared variable with $2M$ degrees of freedom, non-central parameter $2\rho_N$, and weighted by the current est mate of σ_N^2 .

In the noise-only case, the distribution of the summed decision variables is a central chi-squared distribution, written as

$$\sum_{m=1}^{M} u_m = \sigma_N^2 \chi^2(2M) . \quad (8.1-26)$$

A scheme that may provide a workable jammed symbol detection when a signal slot can be jammed is realized by discarding the maximum of the u_m variables which results in the distributional assumption

$$\sum_{m=1}^{M-1} u_m - u_{\max} = \sigma_N^2 \chi^2(2M-2) . \quad (8.1-27)$$

Note that the distribution of (8.1-27) is written as a central chi-square distribution which assumes that the signal itself is always stronger than the jamming. That is, the presence of weak jamming can easily be detected in the absence of the now discarded stronger signal. Should the jamming be strong, we have a situation in which the signal channel is included in the left-over sum. However, this is an acceptable scenario because the signal channel power will help in deciding if a jammed symbol condition exists.

The probability of a false alarm for this scheme is obtained by applying the methodology to formulate the non-signal channel probabilities for ACJ-AGC receivers as discussed in Section 4.3. Specifically, the probability of the sum of the non-signal channel measurement samples being less than the normalized symbol threshold on a given hop is written as

$$\Pr \left\{ \sum u_m < \frac{n_{sym}}{\sigma_N^2} \right\} = 1 - \frac{\Gamma(2M-2; n_{sym}/2\sigma_N^2)}{\Gamma(2M-2)}. \quad (8.1-28)$$

Hence, the probability of a false-alarm for jammed symbol detection is simply

$$P_{FA(sym)} = \frac{\Gamma(2M-2; n_{sym}/2\sigma_N^2)}{\Gamma(2M-2)}. \quad (8.1-29)$$

Another method to implement jammed symbol detection in the presence of a signal would utilize a combination of the individual channel jammed detector outputs. For example, in the case of $M=4$ we can employ the combinatorial logic of any two individual channel detectors' outputs "ANDED" to produce a symbol jamming decision from the six possible combinations. The detection of the "any channel jammed" condition will be the logic variable

$$\begin{aligned} J_{symbol} = & (J_1 \cdot J_2) + (J_1 \cdot J_3) + (J_1 \cdot J_4) \\ & + (J_2 \cdot J_3) + (J_2 \cdot J_4) + (J_3 \cdot J_4) \end{aligned} \quad (8.1-30)$$

where it is assumed that detection of the signal itself has triggered the decision of a jammed channel (J_i) for a particular channel. Thus, we are able to realize a jammed symbol detection when a signal is present. For the case of any M we have the overall false alarm probability (when no signal is present)

$$P_{FA(sym)} = \sum_{n=2}^M \binom{M}{n} P_{FA(CH)}^n (1 - P_{FA(CH)})^{M-n} \quad (8.1-31)$$

which simplifies to

J. S. LEE ASSOCIATES, INC.

$$P_{FA(\text{sym})} \approx \binom{M}{2} P_{FA(\text{CH})}^2 \quad (8.1-32)$$

as the overall false-alarm rate for a jammed symbol detection.

8.2 ASSESSMENT OF IMPLEMENTATION EFFECTS

In this second part of our implementation studies, we investigate the effects of the previously discussed implementation schemes on the performance of the ECCM receivers. We demonstrate the necessary adjustments involved in reformulating the probability of error expressions which are now conditioned on estimated (measured) parameters instead of assumed "perfect" measurement quantities. Our objective is to assess the "return on the investment" realized by resorting to the complex measurement schemes needed to implement the AGC receivers, which for ideal measurements achieve the best ECCM receiver performances in worst-case PBNJ. That is, we seek to answer the question, "Will practical implementations of the AGC receivers continue to outperform the simple SNORM and hard-decision receivers, which require no measurement?"

8.2.1 Methodology for Direct Assessment.

Analyses of the error performance of the implemented AGC schemes are extremely difficult for the following reasons. First, we must account for the measured (estimated) quantities $\hat{\sigma}_N^2$ and $\hat{\sigma}_T^2$ as random variables imbedded in the probability of error expressions. Second, the effect of errors in the JSI decisions upon the $P(e)$ expressions must also be considered. Previous analytical results in this report were derived on the assumption that perfect measurements were obtained for σ_N^2 and σ_T^2 ; this provided a lower bound on the error performance to be realized in practice. This ideal total error probability can be expressed

parametrically by

$$P(e) \equiv P(e; \gamma, \underbrace{E_b/N_0, E_b/N_J, L, M}_{\text{actual parameters}}, \underbrace{\sigma_N^2, \sigma_T^2, S}_{\text{parameters assumed by receivers}}) \quad (8.2-1)$$

and can be written as

$$P(e) = \sum \Pr\{[v]\} P_b(e|[v]), \quad (8.2-2)$$

where $[v]$ is a matrix describing the jamming event (see Section 2.2) over the L-hop diversity. The above expressions must now be restated as,

$$P(e) \equiv P(e; \gamma, E_b/N_0, E_b/N_J, L, M; \hat{\sigma}_N^2, \hat{\sigma}_T^2, \hat{S}) \quad (8.2-3)$$

$$P(e) = \sum \Pr\{[v], [\hat{v}]\} P(e|[v], [\hat{v}]) \quad (8.2-4)$$

in accounting for $\hat{\sigma}_N^2$, $\hat{\sigma}_T^2$, \hat{S} , as well as estimates in JSI.

For example, in the ideal situation the IC-AGC receiver decision variables are

$$z_m = \sum_{k=1}^L z_{mk}; \quad m=1, 2, \dots, M; \quad (8.2-5a)$$

where each z_{mk} is the square-law envelope detector sample x_{mk}^2 multiplied by the weight

$$w_{mk} = \begin{cases} 1/\sigma_N^2, & \text{channel } m \text{ not jammed on hop } k \\ 1/\sigma_T^2, & \text{channel } m \text{ jammed on hop } k. \end{cases} \quad (8.2-5b)$$

J. S. LEE ASSOCIATES, INC.

This ideal normalization results in the $\{z_{mk}\}$ being all unscaled chi-square random variables, as discussed in Section 4.

Now if the a priori quantities σ_N^2 and σ_T^2 are not available and the jamming condition of the channels is not known, we must use estimates $\hat{\sigma}_N^2$ and $\hat{\sigma}_T^2$, and also decide whether the channel is jammed. This results in the weights

$$\hat{w}_{mk} = \begin{cases} (\hat{\sigma}_N^2)^{-1} (1-P_{Fm}) + (\hat{\sigma}_T^2)^{-1} P_{Fm} = W_0, & \text{not jammed;} \\ (\hat{\sigma}_N^2)^{-1} (1-P_{Dm}) + (\hat{\sigma}_T^2)^{-1} P_{Dm} = W_1, & \text{jammed;} \end{cases} \quad (8.2-6)$$

where P_{Fm} and P_{Dm} are per-channel jamming false alarm and detection probabilities. In this description, it is assumed that the variance estimates are developed from look-ahead or in-band measurement data prior to the symbol being processed, and that the channel jamming decision is based on a one-hop look-ahead scheme. In the absence of signals in the look-ahead channels,

$$P_{Fm} = P_F(\hat{\sigma}_N^2, \hat{\sigma}_T^2; \sigma_N^2, \sigma_T^2) \text{ and } P_{Dm} = P_D(\hat{\sigma}_N^2, \hat{\sigma}_T^2; \sigma_N^2, \sigma_T^2);$$

that is, these probabilities are the same for each of the M symbol channels.

Thus, after accumulating the L hops, the decision statistics are, conditioned on jamming events and measurements,

$$\begin{aligned} z_1 &= W_0 \sigma_N^2 \chi^2[2(L-\ell_1); 2(L-\ell_1)\rho_N] \\ &\quad + W_1 \sigma_T^2 \chi^2[2\ell_1; 2\ell_1\rho_T]; \end{aligned} \quad (8.2-7a)$$

$$z_m = W_0 \sigma_N^2 \chi^2[2(L-\ell_m)] + W_1 \sigma_T^2 \chi^2(2\ell_M), \quad m>1; \quad (8.2-7b)$$

J. S. LEE ASSOCIATES, INC.

where $\underline{\ell} = (\ell_1, \ell_2, \dots, \ell_M)$ is the jamming event vector of the number of hops jammed in each channel. Recall now that the linear combining receiver (LCR) has the normalized decision statistics

$$u_1 = z_1/\sigma_N^2 = x^2[2(L-\ell_1); 2(L-\ell_1)\rho_N] + Kx^2[2\ell_1; 2\ell_1\rho_T]; \quad (8.2-8a)$$

$$u_m = z_m/\sigma_N^2 = x^2[2(L-\ell_m)] + Kx^2(2\ell_m), m>1; \quad (8.2-8b)$$

where $K = \sigma_T^2/\sigma_N^2$. We therefore recognize that, conditioned on the measurements, the implemented IC-AGC receiver's BER will have the same functional form as the LCR's with the new K value

$$K' = \frac{\sigma_T^2 w_1}{\sigma_N^2 w_0} = \frac{\sigma_T^2}{\sigma_N^2} \cdot \frac{\hat{\sigma}_T^2(1-P_D) + \hat{\sigma}_N^2 P_D}{\hat{\sigma}_T^2(1-P_F) + \hat{\sigma}_N^2 P_F}. \quad (8.2-9)$$

Evaluation of the effect of the measurements and JSI decisions then involves numerically averaging the LCR error probability (with K replaced by (8.2-9)) over the distributions of $\hat{\sigma}_N^2$ and $\hat{\sigma}_T^2$.

8.2.2 In Search of an Upper Bound: Simplified Measurement Models.

Equations (8.2-7) to (8.2-9) reveal that the implemented IC-AGC receiver statistics more or less tend toward those of the ineffective LCR, depending on the quality of the measurements. This fact underscores the important role of the a priori information utilized by the AGC receivers in

J. S. LEE ASSOCIATES, INC.

their superior performance. We can take the position that the ideal AGC performances calculated in this report represent a lower bound on achievable BER, though perhaps not the lowest bound,* and then seek an upper bound instead of attempting the arduous and time-consuming direct analysis of implemented systems. Such an upper bound, if sufficiently tight, would be suitable for comparison with the BER results for the receivers not employing a priori information.

How shall we obtain an upper bound? Since the performance degradation associated with the receiver implementations is related to the quality of the measurements, we realize that any bound would be directly identified with a particular measurement approach, and parametric in the features of that approach (such as number of samples taken). We fully anticipate, for example, that an upper bound on the implemented system's BER would decrease as the number of samples used in the measurement increases. Therefore, it is reasonable to consider possible implementations utilizing one sample as candidates for systems whose performances represent an upper bound on what is achievable in the same manner as the ideal systems represent a lower bound.

For the ACJ-AGC FH/MFSK receiver, we consider the simplified version that we call the "practical ACJ" (PACJ) receiver. Since the ideal ACJ receiver uses the weights $w_k = (\max_m \sigma_{mk}^2)^{-1}$, we stipulate that the PACJ uses the weights

$$w_k = (\max_m x_{mk}^2). \quad (8.2-10)$$

*In Section 7 it was observed that the SNORM receiver can outperform the AGC receivers in some, limited circumstances.

J. S. LEE ASSOCIATES, INC.

This approach in effect utilizes the received square-law envelope detector samples themselves as (one sample) measurements of the noise power in each channel, and thus is a form of in-band measurement which is very simple indeed compared to schemes discussed in Section 8.1.

We note in passing that this PACJ receiver is related to the hard-decision receiver in the following way: the HD decision statistics $\{z_{mk}\}$ are the PACJ decision statistics after being subjected to a two-level quantization. That is,

$$z_{HD} = \begin{cases} 1, & z_{PACJ} \geq 1 \\ 0, & z_{PACJ} < 1. \end{cases} \quad (8.2-11)$$

For the IC-AGC receiver, we postulate that a one-hop look-ahead implementation yields one detector sample in each of the M channels just prior to the actual symbol's occupancy of those channels.* Since the ideal IC-AGC uses the weights $w_{mk} = (\sigma_{mk}^2)^{-1}$, we can treat the look ahead samples $\{v_{mk}\}$ as one-sample estimates of noise power and specify the "practical IC" (PIC) weights

$$w_{mk} = (v_{mk})^{-1}. \quad (8.2-12)$$

In what follows, we evaluate each of these practical receivers for the case of $M=2$, $L=2$ in order to compare them with their respective ideal receivers. We also can consider the SNORM and hard-decision receivers as

*Alternatively, the hopping and symbol rates could be reduced by one-half in order to sample the channel first, then receive the transmission; look-ahead is avoided in this way, at the expense of data rates.

J. S. LEE ASSOCIATES, INC.

"practical" AGC implementations, and will continue to exhibit their BER results for comparison purposes.

8.2.3 Example Evaluations of Practical AGC Receivers .

We now find the error probabilities for the PACJ and PIC receivers for M=2 and L=2.

8.2.3.1 Analysis of the PACJ receiver.

The PACJ receiver for M=2 is diagrammed in Figure 8.2-1. The decision statistics are

$$z_m = \sum_{k=1}^L \frac{x_{mk}^2}{\max(x_{mk}^2)} , \quad m = 1, 2. \quad (8.2-13)$$

In Appendix C it is shown that this receiver for L=2 and FH/RMFSK has the bit error probability

$$P_b(e) = 2 \int_0^1 dx f(x)G(x) + G^2(1) \quad (8.2-14a)$$

where $K = \sigma_T^2/\sigma_N^2$ and

$$\begin{aligned} f(x) &= \pi_0 \frac{1}{(x+1)^2} \exp \left\{ -\frac{\rho_N x}{(x+1)} \right\} \left[1 + \frac{\rho_N}{x+1} \right] \\ &+ \pi_1 \frac{K}{(Kx+1)^2} \exp \left\{ -\frac{K\rho_T x}{(Kx+1)} \right\} \left[1 + \frac{\rho_T}{Kx+1} \right] \\ &+ \pi_1 \frac{K}{(x+K)^2} \exp \left\{ -\frac{\rho_N x}{(x+K)} \right\} \left[1 + \frac{K\rho_N}{x+K} \right] \\ &+ \pi_2 \frac{1}{(x+1)^2} \exp \left\{ -\frac{\rho_T x}{(x+1)} \right\} \left[1 + \frac{\rho_T}{x+1} \right]. \end{aligned} \quad (8.2-14b)$$

and

$$\begin{aligned} G(x) &= \pi_0 \frac{x}{x+1} e^{-\rho_N/(1+x)} + \pi_2 \frac{x}{x+1} e^{-\rho_T/(1+x)} \\ &+ \pi_1 \frac{Kx}{Kx+1} e^{-\rho_N/(1+Kx)} + \pi_1 \frac{x}{x+K} e^{-K\rho_T/(x+K)}. \end{aligned} \quad (8.2-14c)$$

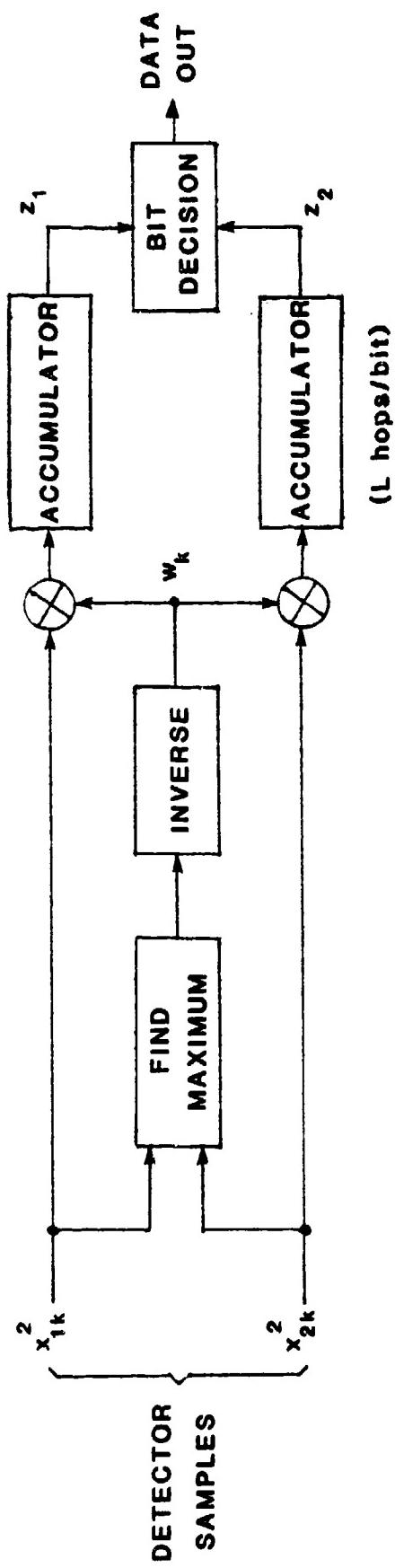


FIGURE 8.2-1 PRACTICAL ACJ-AGC(PACJ) RECEIVER FOR FH/RMFSK WHEN $M=2$

Numerical results for the L=2 PACJ binary receiver are shown for $E_b/N_0 = 13.35$ dB and 20 dB, respectively, in Figures 8.2-2 and 8.2-3. There is no difference between these results and those of the SNORM receiver that can be discerned from the figures - a close look at the data reveals a slight difference except for $\gamma=1$ (full-band jamming), for which analysis shows that the two receivers yield identical performance for the M=2, L=2 case.

8.2.3.2 Analysis of the PIC receiver.

The PIC receiver for M=2 is diagrammed in Figure 8.2-4. The decision statistics are

$$z_m = \sum_{k=1}^L \frac{u_{mk}}{v_{mk}}, \quad m=1,2; \quad (8.2-15)$$

where the $\{u_{mk}\}$ are the usual receiver samples and the $\{v_{mk}\}$ are look-ahead samples. These look-ahead samples are assumed to have the same noise powers as their corresponding "usual" samples, and not to have a signal present.

In Appendix C it is shown that the L=2 FH/RMFSK or FH/MFSK PIC receiver error probability in partial-band noise jamming is given by

$$\begin{aligned} P_b(e) &= (1-\gamma)^2 P_b(e | \rho_1 = \rho_2 = E_b/2N_0) \\ &+ 2\gamma(1-\gamma)P_b(e | \rho_1 = E_b/2N_0, \rho_2 = E_b/2N_T) \\ &+ \gamma^2 P_b(e | \rho_1 = \rho_2 = E_b/2N_T), \end{aligned} \quad (8.2-16a)$$

where γ is the jamming fraction and

$$\begin{aligned} P_b(e | \rho_1, \rho_2) &= \int_0^1 du \int_0^1 dv e^{-\rho_1 u - \rho_2 v} (1+\rho_1 - \rho_2 u)(1+\rho_2 - \rho_2 v) \\ &\times \frac{uv}{u+v} \left[2 + \frac{uv}{u+v} \ln\left(\frac{u+v}{uv} - 1\right) \right]. \end{aligned} \quad (8.2-16b)$$

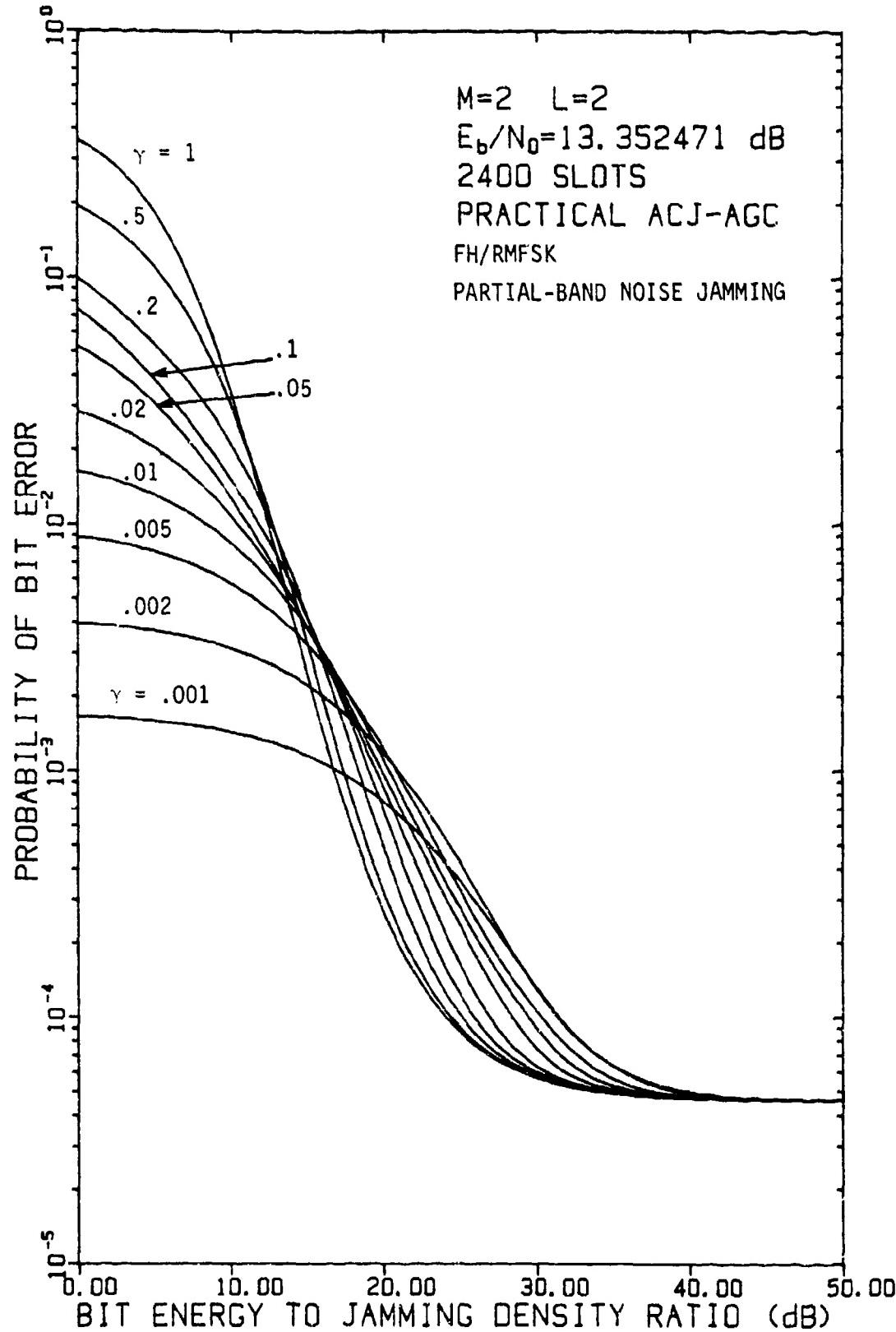


FIGURE 8.2-2 BER PERFORMANCE FOR PACJ FH/RMFSK RECEIVER IN PARTIAL-BAND NOISE JAMMING FOR $M=2$, $L=2$, AND $E_b/N_0 = 13.35 \text{ dB}$

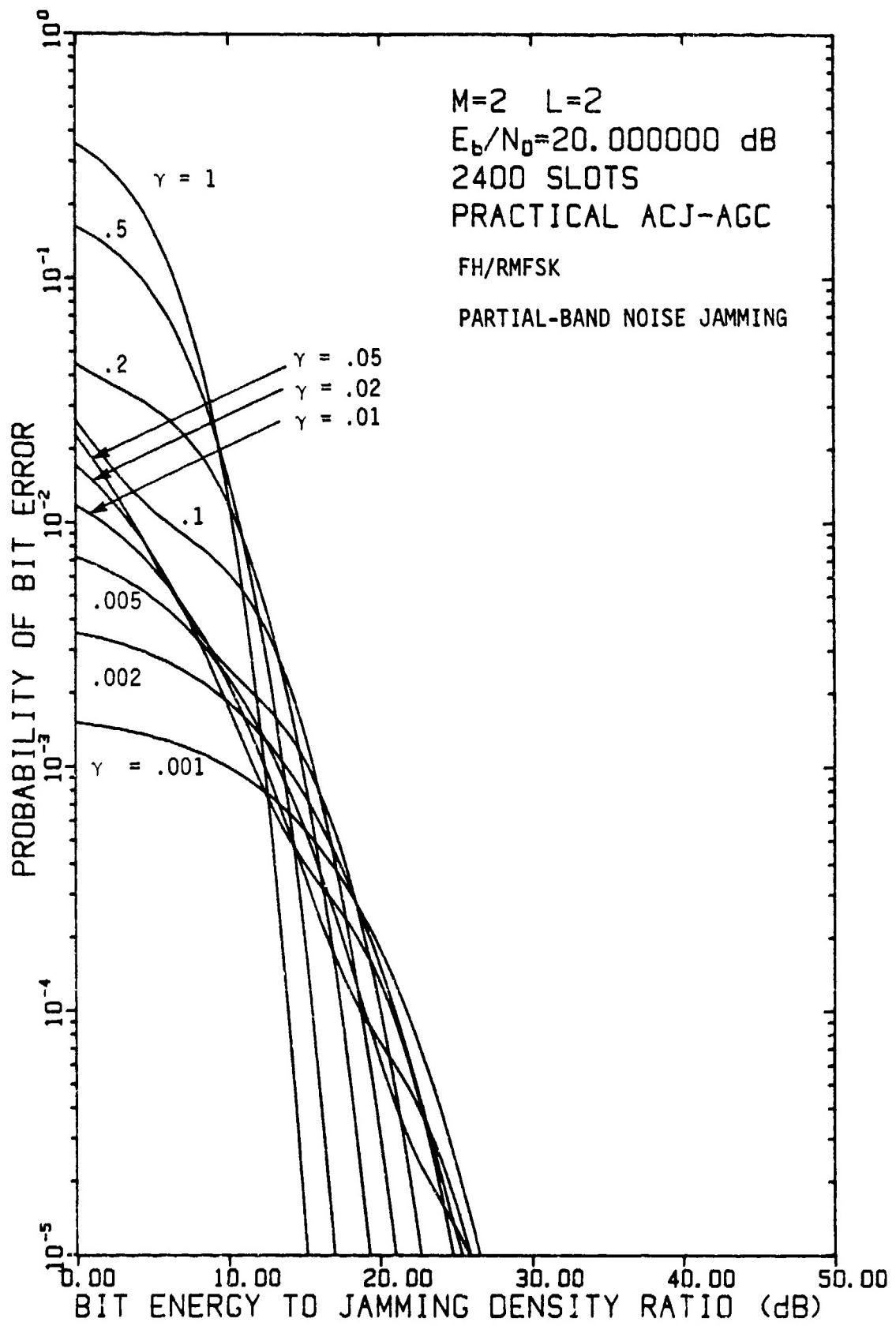


FIGURE 8.2-3 BER PERFORMANCE FOR PACJ FH/RMFSK RECEIVER IN PARTIAL-BAND NOISE JAMMING FOR $M=2$, $L=2$, AND $E_b/N_0 = 20 \text{ dB}$

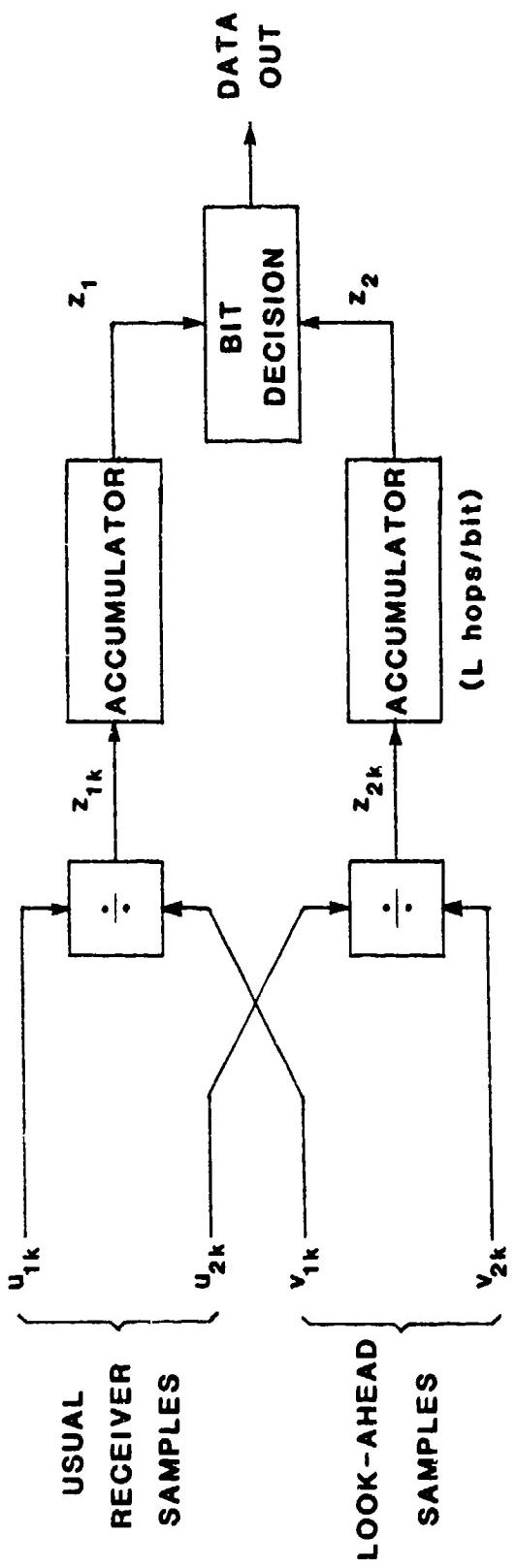


FIGURE 8.2-4 PRACTICAL IC-AGC(PIC) RECEIVER FOR FH/RMF SK WHEN $M=2$

Numerical results for the L=2 PIC binary receiver reveal that it performs very poorly. For example, for full-band noise jamming the BER varies with E_b/N_J as shown in Figure 8.2-5. We note that for high E_b/N_J (practically no jamming) the error probability is greater than 10^{-4} even for thermal noise so small that $E_b/N_0 = 40$ dB. Evidently the predictably poor quality of the one-sample estimate of noise variance is especially damaging when using it in the denominator of the ratios taken in (8.2-15).

With slightly more effort, we find that if the PIC receiver uses two look-ahead noise samples, summed to obtain a better variance estimate, the performance improves considerably. The decision statistics for this version of the PIC receiver are

$$z_m = \sum_{k=1}^L \frac{u_{mk}}{\sqrt{v_{mk1} + v_{mk2}}} , \quad m=1,2. \quad (8.2-17)$$

Using the same analytical approach as in Appendix C, but with the sum of the look-ahead variables being $\sigma_{mk}^2 \chi^2(4)$ distributed, we find that the conditional $P(e)$ for $M=2$ and $L=2$ is

$$\begin{aligned} P_b(e|\rho_1, \rho_2) &= 4 \int_0^1 du \int_0^1 dv (1-u)(1-v) \exp\{-\rho_1(1-u)-\rho_2(1-v)\} \\ &\times [1+2\rho_1^2 u + \rho_1^2 u^2/2] [1+2\rho_2^2 v + \rho_2^2 v^2/2] \\ &\times g\left(\frac{u}{1-u} + \frac{v}{1-v}\right) , \end{aligned} \quad (8.2-18a)$$

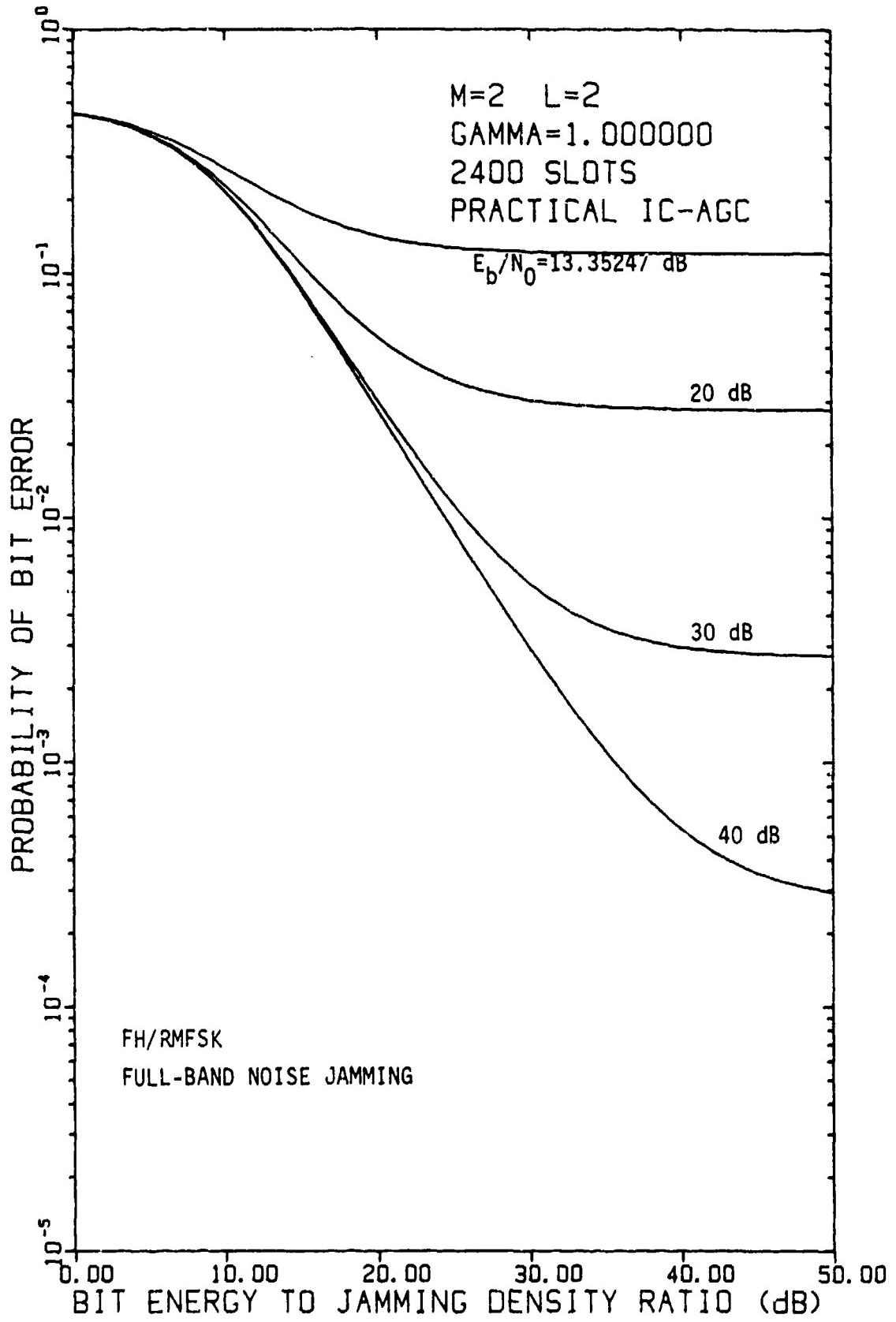


FIGURE 8.2-5 PERFORMANCE OF THE PRACTICAL IC-AGC FH/RMFSK RECEIVER IN FULL-BAND NOISE JAMMING FOR $M=L=2$.

J. S. LEE ASSOCIATES, INC.

where

$$g(x) = \frac{12}{(x+2)^4} \ln(x+1) + \frac{2(x^2+7x+4)}{(x+1)(x+2)^3}. \quad (8.2-18b)$$

Numerical results for full-band jamming are shown in Figure 8.2-6. For $E_b/N_0 = 13.35$ dB, there is not much of an improvement, but for 20 dB and higher E_b/N_0 , there is about a decade improvement over the asymptotic BER obtained using a one-sample noise estimate. Clearly as more look-ahead samples are used, the PIC receiver will act more like the IC-AGC. However, if more measurements are taken, the likelihood increases that the measurement will suffer from changes in the assumed noise environment over the measurement time. Perhaps with the RMFSK hopping scheme it is possible to gather noise samples from adjacent (unused) hopping slots in addition to (or in place of) using a look-ahead approach.

8.3 CONCLUSIONS AND RECOMMENDATIONS

Having considered practical implementations of ECCM receivers for FH/RMFSK in worst-case partial-band noise jamming (WCPBNJ), we are able to conclude our study with some "lessons learned," from which we also can recommend further studies.

8.3.1 Knowledge Gained from Study.

8.3.1.1 Ideal receiver performances.

The ideal receiver performances obtained in Sections 3-6 and compared in Section 7 show for the first time what the expected performance of random frequency-hopping MFSK is in WCPBNJ when L hops per symbol soft decisions are

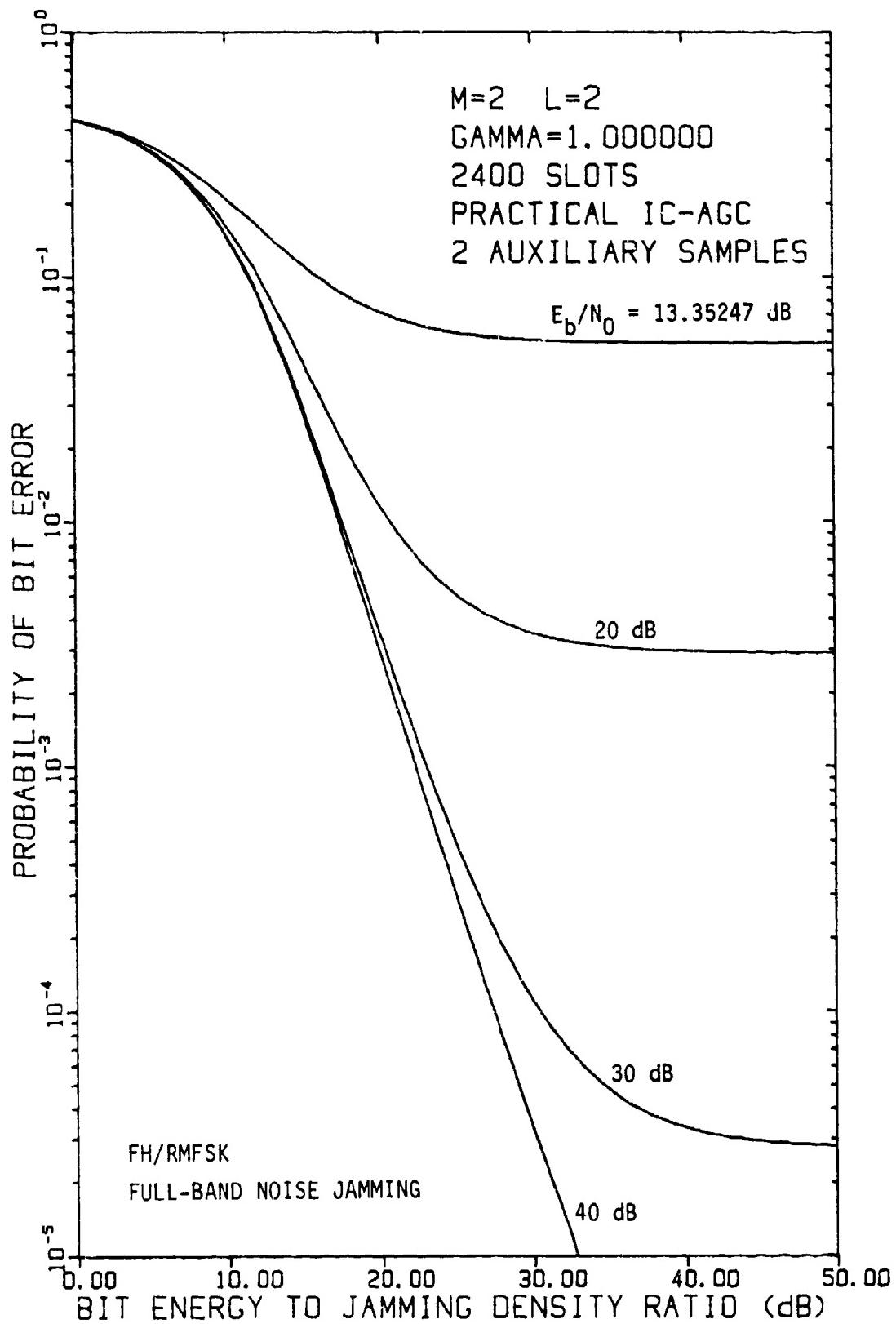


FIGURE 8.2-6 PERFORMANCE OF THE PRACTICAL IC-AGC FH/RMFSK RECEIVER USING A TWO-SAMPLE NOISE ESTIMATE IN FULL-BAND NOISE JAMMING FOR $M=L=2$.

J. S. LEE ASSOCIATES, INC.

used and thermal noise is not neglected. We have learned that per-channel adaptive normalization, such as envisioned using the IC-AGC receiver weighting scheme, is effective in countering the jamming effects which are most damaging to the RMFSK performance: jamming power in non-signal symbol frequency channels and not in the signal channel. The IG-AGC scheme controls the channel gains in such a way as to equalize the a priori noise powers in the channels, in effect forcing the non-Gaussian input noise process to be a Gaussian process. The jamming then only affects the error through the reduction of relative signal power when jammed signal hops are normalized, and RMFSK using this scheme performs the same as the conventional MFSK hopping.

We have learned also that per-symbol adaptive normalization, typified by the ACJ-AGC receiver's weighting scheme, is effective in countering WCPBNJ, though not as effective as per-channel normalization. The per-symbol operation equalizes the maximum of the M channels' a priori noise powers to a constant value, but does not affect the relative powers among the M channels, so that to a certain extent the RMFSK system, unlike MFSK, is still subject to noise power imbalances on each hop and therefore is more vulnerable than MFSK. However, the per-symbol normalization does prevent jammed hops from dominating the soft-decision, and therefore achieves an ECCM or anti-jam diversity effect.

The per-channel and per-symbol AGC schemes perform the same for no jamming, and the performance achieved is sensitive to the amount of thermal and/or background noise present at the receiver, expressed relatively in our

J. S. LEE ASSOCIATES, INC.

study by the ratio E_b/N_0 . For any finite E_b/N_0 value, the receiver performances for weak or no jamming degrade in proportion to L , the number of hops/symbol, due to noncoherent combining losses. It cannot be emphasized too strongly that thermal noise should always be included in any study, because different implementations of the ideal receiver combining schemes in general are subject to different noncoherent combining losses plus any losses due to quantization effects. The ideal AGC performances we have obtained provide a lower bound on achievable performance in the sense that errorless noise power measurements are assumed.

8.3.1.2 Practical receiver performances.

The several "practical" FH/RMFSK receiver combining schemes we have studied may be classified as implementations of either per-channel (IC-AGC) or per-symbol (ACJ-AGC) ideal schemes.

The clipper receiver implements a per-channel ECCM scheme and thus to a certain extent achieves a performance in WCPBNJ whose parametric behavior follows that of the IC-AGC. However, its best clipping threshold value is parametric in received signal and thermal noise powers and in L . In our study we have calculated performances assuming that these powers are known a priori; it is expected that estimation of the correct threshold will degrade its performance. But if we can assume that a reasonably good estimate of un-jammed SNR is available, the clipper receiver appears to be a viable candidate for a practical ECCM receiver.

It was shown that direct implementation of the IC-AGC using auxiliary noise power measurements has the potential for approaching the IC-AGC performance, but only if certain assumptions are made: (a) the receiver complexity can be further advanced economically to include the required noise power measurements (as many as possible per channel); (b) the noise processes

J. S. LEE ASSOCIATES, INC.

being measured are relatively stationary during the time of measurement and not subject to corruption from the signal or other signals. These assumptions are quite restrictive, so that we would choose to implement another, simpler scheme if its performance is satisfactory.

The per-symbol receiver ECCM schemes studied include the self-normalizing receiver (SNORM) and the so-called "practical ACJ" (PACJ). These receiver implementations are very simple, requiring only operations using the usual received envelope samples in the M symbol frequency channels. Somewhat surprisingly, these two schemes perform very well (nearly identically for M=2 and L=2), even better than the supposedly best ideal IC-AGC receiver under certain limited circumstances. Therefore, if the receivers using a priori noise and jamming information tend to represent a lower bound on system performance, and the small-sample size "practical" receivers, an upper bound--then we have observed a situation when lower and upper bounds converge to agree upon a predicted performance result. The implications are that we may regard the easily-calculated IC-AGC performance as representative of achievable system performance, with perhaps a slight implementation loss of a few dB when the simple practical receivers are employed, and when the system noise without jamming is small (high E_b/N_0).

8.3.1.3 RMFSK vs MFSK.

We have found that for smaller alphabet sizes (M=2 or 4), the error performance of FH/RMFSK in worst-case PBNJ is comparable to that of the conventional FH/MFSK, when appropriate receiver processing is employed, to the extent that we state that the price to be paid for the additional RMFSK system

complexity can be assessed against the threat of follow-on noise jamming. That is, if follow-on jamming is not considered a threat, MFSK should be used; but if it is a threat, RMFSK is an effective counter-countermeasure that also works satisfactorily in the worst-case partial-band noise jamming environment.

8.3.2 Recommendations.

With the perspective gained from our study we make the following recommendations for further research.

(a) Derivation of system error performances using the PACJ and similar "nonparametric" ECCM receivers; it is conjectured that analysis would yield BER expressions for $M > 2$ that are more feasible for computation than those for the clipper and self-normalizing receivers we have studied.

(b) Analysis of FH/RMFSK performance under multi-tone jamming; it has been asserted that FH/RMFSK "precludes" systematic tone jamming, but what, in quantitative terms, is its vulnerability to tone jamming, relative to FH/MFSK?

(c) Analysis of mutual interference effects in an FH/RMFSK system; these are considered to be more numerous than for FH/MFSK, and possibly more damaging - a more intricate setup for multiple users may be necessary to avoid mutual interference.

APPENDIX A

PROBABILITY DENSITY FUNCTIONS FOR SOFT DECISION
RECEIVER STATISTICS

For soft decision receivers with multiple hops per symbol, the M decision statistics are of the form

$$z_m = \sigma_{1m}^2 \chi^2(v_{1m}; \lambda_{1m}) + \sigma_{2m}^2 \chi^2(v_{2m}; \lambda_{2m}), \quad (A-1)$$

where $\chi^2(v; \lambda)$ denotes a noncentral chi-squared random variable with v degrees of freedom and noncentrality parameter λ , and σ_{1m}^2 and σ_{2m}^2 are different scalings. For the cases to be studied we can also write

$$u_m = z_m / \sigma_{1m}^2 = \chi^2(2L - 2\ell_m; \lambda_{1m}) + K_m \chi^2(2\ell_m; \lambda_{2m}), \quad (A-2)$$

since $v_{1m} + v_{2m} = 2L$, twice the number of hops per symbol, and $v_{2m} = 2\ell_m$, twice the number of jammed hops for symbol channel m . Also, without loss of generality we designate the first ($m = 1$) channel as the one containing the transmitted signal, and the others ($m \geq 2$) as containing noise only. Thus

$$\lambda_{11} = 2(L - \ell_1)S/\sigma_{11}^2, \quad \lambda_{21} = 2\ell_1 S/\sigma_{21}^2 \quad (A-3a)$$

and

$$\lambda_{1m} = \lambda_{2m} = 0, \quad m \geq 2. \quad (A-3b)$$

Previously it has been shown [1, Appendix 2E] that the probability density function (pdf) for u_m given by (A-2) is $p_u(u; \ell_m, \lambda_{1m}, \lambda_{2m}, K_m)$, where

J. S. LEE ASSOCIATES, INC.

$$p_u(\alpha; \ell, \lambda_1, \lambda_2, K)$$

$$= \frac{K-\ell}{2} \exp\left(-\frac{\lambda_1+\lambda_2+\alpha}{2}\right) \sum_{k=0}^{\infty} \sum_{r=0}^{\infty} \frac{\left(\frac{\lambda_1}{2}\right)^k \left(\frac{\lambda_2}{2K}\right)^r \left(\frac{\alpha}{2}\right)^{k+r+L-1}}{k! r! (k+r+L-1)!}$$

$$\times {}_1F_1(r+\ell; k+r+L; \frac{K-1}{K} + \frac{\alpha}{2}), \quad (A-4)$$

where ${}_1F_1(a; b; x)$ is the confluent hypergeometric function. The computation of this expression is very time consuming; in this Appendix we consider alternative expressions that can be computed more quickly, or perhaps be amenable to approximations.

By expanding the confluent hypergeometric function in its series form,

$${}_1F_1(r+\ell; k+r+L; \frac{K-1}{K} + \frac{\alpha}{2}) \quad (A-5)$$

$$= \sum_{n=0}^{\infty} \left(\frac{K-1}{K} + \frac{\alpha}{2} \right)^n \frac{1}{n!} \frac{(r+\ell)_n}{(k+r+L)_n},$$

with

$$(a)_n = \Gamma(a+n)/\Gamma(a), \quad (A-6)$$

and summing over the index k , we obtain the expression

$$\begin{aligned} p_u(\alpha) &= \frac{K-\ell}{2} \exp\left(-\frac{\lambda_1+\lambda_2+\alpha}{2}\right) \sum_{n=0}^{\infty} \left(\frac{K-1}{K}\right)^n \frac{1}{n!} \\ &\times \sum_{r=0}^{\infty} \frac{\left(\frac{\lambda_2}{2K}\right)^r}{r!} (r+\ell)_n \left(\frac{\alpha}{\lambda_1}\right)^{(r+n+L-1)/2} I_{r+m+L-1}(\sqrt{\alpha\lambda_1}) \end{aligned} \quad (A-7)$$

J. S. LEE ASSOCIATES, INC.

This form is based on recognizing the series for $I_{r+n+L-1}(x)$, the modified Bessel function of the first kind of order $r+n+L-1$:

$$I_{r+n+L-1}(x) = \sum_{k=0}^{\infty} \frac{(x/2)^{2k+r+n+L-1}}{k! (k+r+n+L-1)!} . \quad (A-8)$$

Further, we recognize that

$$\left(\frac{\alpha}{\lambda_1}\right)^{(q-1)/2} I_{q-1}(\sqrt{\alpha\lambda_1}) = 2e^{(\alpha+\lambda_1)/2} p_{\chi^2}(\alpha; 2q, \lambda_1), \quad (A-9)$$

where $p_{\chi^2}(\alpha; v, \lambda)$ is the pdf for a noncentral chi-squared random variable with v degrees of freedom and noncentrality parameter λ . This allows us to write

$$p_u(\alpha) = K^{-\ell} e^{-\lambda_2/2} \sum_{n=0}^{\infty} \frac{\left(\frac{K-1}{K}\right)^n}{n!} \sum_{r=0}^{\infty} \frac{\left(\frac{\lambda_2}{2K}\right)^r}{r!} (r+\ell)_n \\ \times p_{\chi^2}(\alpha; 2r+2n+2L; \lambda_1) \quad (A-10)$$

Now we concentrate on the summation over the indices r and n .

Since

$$\sum_{n=0}^{\infty} \sum_{r=0}^{\infty} f(n, r) g(n + r) \\ = \sum_{n=0}^{\infty} g(n) \sum_{r=0}^n f(n-r, r), \quad (A-11)$$

J. S. LEE ASSOCIATES, INC.

we can express the pdf in the form

$$p_u(\alpha) = \sum_{n=0}^{\infty} c_n p_{X^2}(\alpha; 2L+2n, \lambda_1), \quad (A-12)$$

where

$$c_n = e^{-\lambda_2/2} \left(\frac{K-1}{K}\right)^n \frac{1}{K^\ell} \sum_{r=0}^n \frac{\left(\frac{\lambda_2/2}{K-1}\right)^r}{r!} \frac{(r+\ell)_{n-r}}{(n-r)!}. \quad (A-13)$$

Now,

$$\begin{aligned} \frac{(r+\ell)_{n-r}}{(n-r)!} &= \frac{(r+\ell+n-r-1)!}{(r+\ell-1)!(n-r)!} \\ &= \binom{n+\ell-1}{n-r}; \end{aligned} \quad (A-14)$$

and

$$\sum_{r=0}^n \binom{n+a}{n-r} \frac{(-x)^r}{r!} = \mathcal{L}_n^a(x), \quad (A-15)$$

with $\mathcal{L}_n^a(x)$ the generalized Laguerre polynomial. Thus,

$$c_n = e^{-\lambda_2/2} \left(\frac{K-1}{K}\right)^n \frac{1}{K^\ell} \mathcal{L}_n^{\ell-1} \left[-\frac{\lambda_2/2}{K-1}\right]. \quad (A-16)$$

J. S. LEE ASSOCIATES, INC.

Since (A-12) is a pdf, it must integrate over α to unity. This requires that

$$\sum_{n=0}^{\infty} c_n = 1. \quad (A-17)$$

In fact, this is so, since [3, eq. 8.975.1]

$$\begin{aligned} \sum_{n=0}^{\infty} b^n \ell_n^a(x) &= (1-b)^{-a-1} \exp \left\{ \frac{bx}{b-1} \right\} \\ &= K^\ell e^{\lambda_2/2}. \end{aligned} \quad (A-18)$$

Approximation

The expression (A-12) for the pdf is in the form of a series of weighted chi-squared pdf's. This suggests an approximation based on truncating the series:

$$p_u(\alpha) \approx \sum_{n=0}^N c_n p_{\chi^2}(\alpha; 2L+2n, \lambda_1) / \sum_{n=0}^N c_n. \quad (A-19)$$

Since, even for $\lambda_2 = 0$,

$$\frac{c_{n+1}}{c_n} = \frac{1}{n} \frac{\binom{K-1}{K}}{1-1/n} \cdot \frac{1+\ell/n}{1+(\ell-1)/n} \rightarrow \frac{\binom{K-1}{K}}{n} < \frac{1}{n}, \quad (A-20)$$

the truncation is feasible but, depending on the value of K , the convergence of the weights $\{c_n\}$ is slow.

J. S. LEE ASSOCIATES, INC.

APPENDIX B

COMBINATORIAL RELATIONS FOR JAMMING EVENT ENUMERATION

8.1 NUMBERS OF ORDERED VECTORS

We define

$$\begin{aligned} (L+1)S_M(L) &\triangleq \#\{\underline{\ell}: \underline{\ell} = (\ell_1; \ell_2 \leq \ell_3 \leq \dots \leq \ell_M)\} \\ &= (L-1) \sum_{\ell_M=0}^L \sum_{\ell_{M-1}=0}^{\ell_M} \dots \sum_{\ell_2=0}^{\ell_3} (1). \end{aligned} \quad (B.1-1)$$

By direct manipulation,

$$S_2(L) = \sum_{\ell_2=0}^L 1 = L+1 = \binom{L+1}{1} \quad (B.1-2)$$

$$\begin{aligned} S_3(L) &= \sum_{\ell_3=0}^L \sum_{\ell_2=0}^{\ell_3} 1 = \sum_{\ell_3=0}^L S_2(\ell_3) \\ &= \sum_{\ell_3=0}^L \binom{\ell_3 + 1}{1} = \binom{L+2}{2}, \end{aligned} \quad (B.1-3)$$

using [3, equation 0.151.1]. Assuming that

$$S_M(L) = \binom{L+M-1}{M-1}, \quad (B.1-4)$$

we find that

$$\begin{aligned} S_{M+1}(L) &= \sum_{\ell_{M+1}=0}^L S_M(\ell_{M+1}) \\ &= \sum_{\ell_{M+1}=0}^L \binom{\ell_{M+1} + M-1}{M-1} = \binom{L+M}{M}. \end{aligned} \quad (B.1-5)$$

Thus (B.1-4) is proved by induction.

J. S. LEE ASSOCIATES, INC.

B.2 NUMBERS OF PARTITIONED ORDERED VECTORS

We define $R_M(L;n)$ as $(L+1)^{-1}$ times the number of ordered vectors $\underline{\ell}$ such that there are exactly n partitions of the components ($\ell_2 \leq \ell_3 \leq \dots \leq \ell_M$) such that the ℓ_m in the partition are equal. For example,

$$\begin{aligned} R_M(L;1) &= \sum_{\ell_M=0}^L \sum_{\ell_{M-1}=0}^{\ell_M} \dots \sum_{\ell_2=0}^{\ell_3} U(\ell_2 = \ell_3 = \ell_4 = \dots = \ell_M) \\ &= \sum_{\ell_M=0}^L 1 = (L+1), \end{aligned} \quad (\text{B.2-1})$$

where $U(\cdot)$ is 1 if the relation in the argument is true and zero otherwise.

$R_M(L;1)$ is the number of ordered vectors where the components are all equal.

For two partitions, there are many cases, but they all produce the sum

$$R_M(L;2) = \sum_{r_2=1}^L \sum_{r_1=0}^{r_2-1} 1 = \sum_{r_2=0}^L r_2 = \binom{L+1}{2}, \quad (\text{B.2-2})$$

from [3, equation 0.121.1]. We find that

$$R_M(L;n) = \sum_{r_n=0}^L R_{M-1}(r_n-1; n-1). \quad (\text{B.2-3})$$

Asserting that

$$R_M(L;n) = \binom{L+1}{n} \equiv R(L;n) \quad (\text{B.2-4})$$

and substituting (B.2-4) in (B.2-3) establishes this relation by inductive proof.

J. S. LEE ASSOCIATES, INC.

B.3 TOTAL NUMBER OF VECTORS REPRESENTED BY ORDERED VECTORS

Since there are

$$\binom{M-1}{n_0, n_1, \dots, n_L} \quad (B.3-1)$$

$\underline{\ell}$ vectors represented by each $\underline{\ell}'$ vector (see Section 2.2), the total number of vectors represented is $(L+1)$ times the summation

$$\sum_{\ell_M=0}^L \sum_{\ell_{M-1}=0}^{\ell_M} \cdots \sum_{\ell_2=0}^{\ell_3} \binom{M-1}{n_0, n_1, \dots, n_L} \triangleq T_M(L). \quad (B.3-2)$$

Using the partitioning relations, we can write

$$\begin{aligned} T_M(L) &= \sum \binom{M-1}{n_0, n_1, \dots, n_L} \times \left(\begin{array}{l} \text{number of partitioned } \underline{\ell}' \\ \text{which produce } n_0, n_1, \dots, n_L \end{array} \right) \\ &= \sum_{n=1}^{M-1} R(L;n) \sum_{\text{partitions}} \binom{M-1}{q_1, q_2, \dots, q_n} \cdot \binom{n}{r_1, r_2, \dots, r_{M-1}}, \end{aligned} \quad (B.3-3)$$

where

$$q_k \triangleq \text{number of equal } \underline{\ell}'\text{'s in partition } k \quad (B.3-4a)$$

$$r_s \triangleq \text{number of } q_k \text{ equal to } s. \quad (B.3-4b)$$

Thus

$$T_2(L) = \sum_{n=1}^1 R(L;n) \sum_{\text{part.}} \binom{1}{q_1, q_2, \dots, q_n} \binom{n}{r_1} = L+1, \quad (B.3-5)$$

J. S. LEE ASSOCIATES, INC.

and

$$\begin{aligned}
 T_3(L) &= \sum_{n=1}^2 R(L;n) \sum_{\text{part.}} \binom{2}{q_1, q_2, \dots, q_n} \binom{n}{r_1, r_2} \\
 &= \binom{L+1}{1} \binom{2}{2} \binom{1}{1} + \binom{L+1}{2} \binom{2}{1, 1} \binom{2}{2, 0} \\
 &= L + 1 + 2 \binom{L+1}{2} = (L+1)^2
 \end{aligned} \tag{B.3-6}$$

Calculations show that

$$\begin{aligned}
 T_4(L) &= (L+1)^3 \\
 T_5(L) &= (L+1)^4 \\
 \text{and } T_6(L) &= (L+1)^5.
 \end{aligned}$$

It can be shown [4, p. 106] that

$$T_M(L) = (L+1)^{M-1}, \tag{B.3-7}$$

giving the total of $(L+1)^M$ vectors.

J. S. LEE ASSOCIATES, INC.

B.4 NUMBERS OF ACJ-AGC JAMMING EVENTS

From Section 4, the jamming events are described by the vector $\underline{\ell}$ and the number of hops with at least one channel jammed, ℓ_0 .

B.4.1 Number of $\{\ell_0, \underline{\ell}\}$ events

For a given ℓ_0 and $\underline{\ell}$, the number of events may be counted directly using

$$\#(\ell_0, \underline{\ell}) = \binom{L}{\ell_0} \sum_{v_1 > 0} \dots \sum_{v_{\ell_0} > 0} \delta\left(\sum_{k=1}^{\ell_0} v_k, \underline{\ell}\right). \quad (\text{B.4-1})$$

Previously we have established that the number of $\underline{\ell}$ vectors for n hops is

$$\mathcal{S}(n) \triangleq \sum_{v_1} \dots \sum_{v_n} \delta\left(\sum_{k=1}^n v_k, \underline{\ell}\right) \prod_{m=1}^M \binom{n}{\ell_m} \quad (\text{B.4-2})$$

Note that this quantity is zero for $n < \ell_X = \max_m \ell_m$. For notational convenience, let the sum in (B.4-1) be represented as

$$S(\ell_0) = \sum_{[v:\ell_0] > 0} \delta(\cdot, \cdot) \equiv (\text{sum over all } [v] \text{ with } \ell_0 \text{ non-zero columns and } M \text{ rows}). \quad (\text{B.4-3})$$

Then we find that

$$\sum_{[v:\ell_0] > 0} = \sum_{[v:\ell_0]} - \sum_{[v:\ell_0]} \text{(at least one zero column of } [v:\ell_0]), \quad (\text{B.4-4a})$$

or

$$S(\ell_0) = \mathcal{S}(\ell_0) - (\text{sums with at least one zero column}). \quad (\text{B.4-4b})$$

J. S. LEE ASSOCIATES, INC.

Now, a sum with exactly n non-zero columns of $[v]$ is equivalent to $S(n)$, so that

$$S(\ell_0) = \mathcal{S}(\ell_0) - \sum_{n=1}^{\ell_0} \binom{\ell_0}{n} S(\ell_0-n). \quad (\text{B.4-5})$$

For example,

$$S(1) = \mathcal{S}(1) - S(0) = \mathcal{S}(1) - S(0) \quad (\text{B.4-6a})$$

$$\begin{aligned} S(2) &= \mathcal{S}(2) - 2S(1) - S(0) \\ &= \mathcal{S}(2) - 2\mathcal{S}(1) + \mathcal{S}(0) \end{aligned} \quad (\text{B.4-6b})$$

$$\begin{aligned} S(3) &= \mathcal{S}(3) - 3S(2) - 3S(1) - S(0) \\ &= \mathcal{S}(3) - 3\mathcal{S}(2) - 3\mathcal{S}(1) - \mathcal{S}(0). \end{aligned} \quad (\text{B.4-6c})$$

From these examples we conjecture that

$$S(n) = \sum_{k=0}^n \binom{n}{k} (-1)^k \mathcal{S}(n-k); \quad (\text{B.4-7})$$

substitution of (B.4-7) into (B.4-5) for $\ell_0=n+1$ leads to an inductive proof.

Therefore, we obtain the result

$$\begin{aligned} \#(\ell_0, \underline{\ell}) &= S(\ell_0) \\ &= \binom{\underline{\ell}}{\ell_0} \sum_{r=0}^{\ell_0} \binom{\ell_0}{r} (-1)^r \prod_{m=1}^M \binom{\ell_0-r}{\underline{\ell}_m} \end{aligned} \quad (\text{B.4-8a})$$

$$= \binom{\underline{\ell}}{\ell_0} \sum_{r=0}^{\ell_0-\ell_X} \binom{\ell_0}{r} (-1)^r \prod_{m=1}^M \binom{\ell_0-r}{\underline{\ell}_m}, \quad (\text{B.4-8b})$$

since terms of the sum are zero for $r > \ell_0 - \ell_X$.

J. S. LEE ASSOCIATES, INC.

B.4.2 Summation over $\underline{\lambda}_0$ events

We now demonstrate that summation of (B.4-8a) over $\underline{\lambda}_0$ gives the total number of $\underline{\lambda}$ vectors. We use the fact that

$$\binom{\underline{\lambda}_0 - r}{k} = \frac{1}{k!} \left. \frac{\partial^k}{\partial x^k} (1+x)^{\underline{\lambda}_0 - r} \right|_{x=0}. \quad (\text{B.4-9})$$

Substituting this M times (once for each $\underline{\lambda}_m = k$) yields

$$\begin{aligned} \sum_{\underline{\lambda}_0=0}^L S(\underline{\lambda}_0) &= \left\{ \left(\prod_{m=1}^M \frac{1}{\underline{\lambda}_m!} \left. \frac{\partial^{\underline{\lambda}_m}}{\partial x_m^{\underline{\lambda}_m}} (1+x_m)^{\underline{\lambda}_0} \right|_{x=0} \right) \sum_{\underline{\lambda}_0=0}^L \binom{L}{\underline{\lambda}_0} \sum_{r=0}^{\underline{\lambda}_0} \binom{\underline{\lambda}_0}{r} (-1)^r \right. \\ &\quad \cdot \left. \left[\prod (1+x_m) \right]^{\underline{\lambda}_0 - r} \right\}_{x=0} \\ &= \left\{ \prod_{m=1}^M \frac{1}{\underline{\lambda}_m!} \left. \frac{\partial^{\underline{\lambda}_m}}{\partial x_m^{\underline{\lambda}_m}} (1+x_m)^L \right|_{x=0} \right\}_{x=0} = \prod_{m=1}^M \binom{L}{\underline{\lambda}_m}. \end{aligned} \quad (\text{B.4-10})$$

B.4.3 Summation over $\underline{\lambda}$ events

The number of $\underline{\lambda}_0$ events can be found by summing (B.4-8a) over all possible $\underline{\lambda}$ vectors. This is found to be

$$\begin{aligned} \#(\underline{\lambda}_0) &= \binom{L}{\underline{\lambda}_0} \sum_{r=0}^{\underline{\lambda}_0} \binom{\underline{\lambda}_0}{r} (-1)^r \sum_{\underline{\lambda}} \prod_{m=1}^M \binom{\underline{\lambda}_0 - r}{\underline{\lambda}_m} \\ &= \binom{L}{\underline{\lambda}_0} \sum_{r=0}^{\underline{\lambda}_0} \binom{\underline{\lambda}_0}{r} (-1)^r (\underline{\lambda}_0 - r + 1)^M. \end{aligned} \quad (\text{B.4-11})$$

J. S. LEE ASSOCIATES, INC.

For example,

$$\#(\ell_0=0) = 1 \quad (\text{B.4-12})$$

$$\#(\ell_0=1) = L (2^M - 1) \quad (\text{B.4-13})$$

$$\#(\ell_0=2) = \binom{L}{2} (3^M - 2 \cdot 2^M + 1) \quad (\text{B.4-14})$$

$$\#(\ell_0=L) = (L+1)^M - L \cdot L^M + \binom{L}{2} (L-1)^M + \dots \quad (\text{B.4-15})$$

APPENDIX C

DERIVATION OF ERROR RATE EXPRESSIONS FOR "PRACTICAL ACJ" AND "PRACTICAL IC" RECEIVERS

C.1 JOINT PDF FOR ONE PAIR OF SAMPLES (PACJ).

The error expression will be obtained for M=2 and L=2. For a single pair of square-law envelope detector samples the joint pdf is

$$p_0(x_1, x_2) = c_1 c_2 e^{-c_1 x_1 - c_2 x_2 - \rho_1} I_0(2\sqrt{\rho_1 c_1 x_1}), \quad (C.1-1a)$$

where

$$c_i = \begin{cases} 1/2\sigma_N^2, & \text{channel unjammed;} \\ 1/2\sigma_f^2, & \text{channel jammed,} \end{cases} \quad (C.1-1b)$$

and the signal is assumed to be in channel 1. The normalized variables (z_1, z_2) resulting from this pair have the pdf

$$p_1(z_1, z_2) = p_a(z_1, z_2; x_1 > x_2) + p_b(z_1, z_2; x_1 < x_2). \quad (C.1-2)$$

Now, when $x_1 > x_2$, z_1 is made equal to 1 and $z_2 = x_2/x_1$; thus

$$p_a(z_1, z_2; x_1 > x_2) = \delta(z_1 - 1) \int_0^\infty d\zeta \zeta p_0(\zeta, \zeta z_2), \quad 0 \leq z_2 \leq 1. \quad (C.1-3)$$

Similarly, when $x_2 < x_1$,

$$p_b(z_1, z_2; x_1 < x_2) = \delta(z_2 - 1) \int_0^\infty d\zeta \zeta p_0(\zeta z_1, \zeta), \quad 0 \leq z_1 \leq 1. \quad (C.1-4)$$

The result is that z_1 and z_2 , a single pair of normalized variables, have the joint pdf (conditioned on the possible jamming events) given by

J. S. LEE ASSOCIATES, INC.

$$p_1(z_1, z_2; c_1, c_2) = \frac{c_1 c_2}{(c_1 z_1 + c_2 z_2)^2} \exp \left\{ -\frac{c_2 z_2 \rho_1}{c_1 z_1 + c_2 z_2} \right\} \left[1 + \frac{\rho_1 c_1 z_1}{c_1 z_1 + c_2 z_2} \right]$$

$$\times [\delta(z_1 - 1) + \delta(z_2 - 1)], \quad 0 \leq z_1, z_2 \leq 1. \quad (C.1-5)$$

By direct integration it may be shown that

$$\Pr\{z_1 < z_2\} = \frac{c_1}{c_1 + c_2} \exp \left\{ -\frac{\rho_1 c_2}{c_1 + c_2} \right\} \quad (C.1-6)$$

and that

$$\Pr\{z_1 > z_2\} = 1 - \Pr\{z_1 < z_2\}; \quad (C.1-6)$$

thus the pdf integrates to unity as required.

Taking into account the four possible jamming events, the unconditional pdf may be written using

$$f(z_1, z_2) = \pi_0 \cdot \frac{1}{(z_1 + z_2)^2} \exp \left\{ -\frac{\rho_N z_2}{z_1 + z_2} \right\} \left[1 + \frac{\rho_N z_1}{z_1 + z_2} \right]$$

$$+ \pi_1 \cdot \frac{\kappa}{(z_1 + \kappa z_2)^2} \exp \left\{ -\frac{\kappa \rho_T z_2}{z_1 + \kappa z_2} \right\} \left[1 + \frac{\kappa \rho_T z_1}{z_1 + \kappa z_2} \right]$$

$$+ \pi_1 \cdot \frac{\kappa}{(\kappa z_1 + z_2)^2} \exp \left\{ -\frac{\kappa \rho_N z_2}{\kappa z_1 + z_2} \right\} \left[1 + \frac{\kappa \rho_N z_1}{\kappa z_1 + z_2} \right]$$

$$+ \pi_2 \cdot \frac{1}{(z_1 + z_2)^2} \exp \left\{ -\frac{\rho_T z_2}{z_1 + z_2} \right\} \left[1 + \frac{\rho_T z_1}{z_1 + z_2} \right];$$

$$0 \leq z_1, z_2 \leq 1. \quad (C.1-7)$$

J. S. LEE ASSOCIATES, INC.

With this function the unconditioned pdf becomes

$$p_1(z_1, z_2) = f(z_1, z_2) [\delta(z_1-1) + \delta(z_2-1)]. \quad (\text{C.1-8})$$

C.2 JOINT PDF FOR SUMS OF TWO SAMPLES (PACJ).

Using convolution, the joint pdf for L=2 is

$$p_2(z_1, z_2) = \int_{\max(0, z_1-1)}^{\min(1, z_1)} dv_1 \int_{\max(0, z_2-1)}^{\min(1, z_2)} dv_2 p_1(v_1, v_2) p_1(z_1-v_1, z_2-v_2), \quad (\text{C.2-1})$$

which reduces to

$$\begin{aligned} p_2(z_1, z_2) &= \delta(z_1-2) \int_{\max(0, z_2-1)}^{\min(1, z_2)} dv_2 f(1, v_2) f(1, z_2-v_2) \\ &\quad + 2f(1, z_2-1)f(z_1-1, 1)u(z_2-1)u(z_1-1) \\ &\quad + \delta(z_2-2) \int_{\max(0, z_1-1)}^{\min(1, z_1)} dv_1 f(v_1, 1) f(z_1-v_1, 1), \\ &\quad 0 \leq z_1, z_2 \leq 2. \end{aligned} \quad (\text{C.2-2})$$

C.3 ERROR PROBABILITY FOR L=2 (PACJ).

The error probability is, using (C.2-2),

$$\begin{aligned} P(e) = \Pr\{z_2 > z_1\} &= 2 \int_1^2 dz_2 \int_1^{z_2} dz_1 f(1, z_2-1) f(z_1-1, 1) \\ &\quad + \int_0^1 dz_2 \int_0^{z_2} dv_1 f(v_1, 1) f(z_1-v_1, 1) \\ &\quad + \int_1^2 dz_2 \int_{z_2-1}^1 dv_1 f(v_1, 1) f(z_1-v_1, 1). \end{aligned} \quad (\text{C.3-1})$$

J. S. LEE ASSOCIATES, INC.

Manipulation of the integrals yields

$$\begin{aligned}
 P(e) &= 2 \int_0^1 dx f(1,x) \int_0^x dy f(y,1) \\
 &\quad + \int_0^1 dx \int_0^x dy f(y,1) f(x-y,1) \\
 &\quad + \int_0^1 dx \int_x^1 dy f(y,1) f(x+1-y,1) \tag{C.3-2}
 \end{aligned}$$

$$\begin{aligned}
 &= 2 \int_0^1 dx f(1,x) G(x) \\
 &\quad + \int_0^1 dy f(y,1) G(1-y) \\
 &\quad + \int_0^1 dy f(y,1) [G(1) - G(1-y)] \tag{C.3-3}
 \end{aligned}$$

$$= 2 \int_0^1 dx f(1,x) G(x) + G^2(1), \tag{C.3-4}$$

where $G(x)$ is defined as

$$G(x) \triangleq \int_0^x du f(u,1) \tag{C.3-5}$$

$$\begin{aligned}
 &= \pi_0 \cdot \frac{x}{1+x} \exp \left\{ -\frac{\rho_N}{1+x} \right\} \\
 &\quad + \pi_1 \cdot \frac{x}{K+x} \exp \left\{ -\frac{K\rho_T}{K+x} \right\} \\
 &\quad + \pi_1 \cdot \frac{Kx}{1+Kx} \exp \left\{ -\frac{\rho_N}{1+Kx} \right\} \\
 &\quad + \pi_2 \cdot \frac{x}{1+x} \exp \left\{ -\frac{\rho_T}{1+x} \right\}. \tag{C.3-6}
 \end{aligned}$$

J. S. LEE ASSOCIATES, INC.

Analytically it can be shown that, for $\pi_0 = 1$ (no jamming) or $\pi_2 = 1$ (full-band jamming), the $P(e)$ equals

$$P(e) = \frac{1}{2} e^{-\rho} (1 + \rho/3), \quad (C.3-7)$$

where $\rho = \frac{1}{2} E_b/N_0$ for $\pi_0 = 1$ and $\rho = \frac{1}{2} E_b/N_T$ for $\pi_2 = 1$. This is precisely the same performance obtained by the self-normalizing receiver in Section 5, for the same conditions.

J. S. LEE ASSOCIATES, INC.

C.4 JOINT PDF FOR ONE PAIR OF SAMPLES (PIC).

The joint pdf of the usual and the look-ahead receiver square-law envelope detector samples is

$$p_0(u_1, u_2, v_1, v_2) = c_1^2 c_2^2 e^{-c_1(u_1+v_1)-c_2(u_2+v_2)} I_0(2\sqrt{\rho_1 c_1 u_1}), \quad (C.4-1)$$

where the constants are defined in (C.1-1b). The normalized variables are

$z_{1k} = u_1/v_1$ and $z_{2k} = u_2/v_2$, with the joint pdf

$$\begin{aligned} p_{z_{1k}, z_{2k}}(\alpha, \beta) &= \int_0^\infty dv_1 \int_0^\infty dv_2 v_1 v_2 p_0(v_1^\alpha, v_2^\beta, v_1, v_2) \\ &= \frac{1}{(1+\beta)^2} \cdot \frac{1}{(1+\alpha)^2} \cdot \exp\left\{-\frac{\rho_1}{1+\alpha}\right\} \left[1 + \frac{\frac{\rho_1 \alpha}{1+\alpha}}{1+\alpha}\right] \end{aligned} \quad (C.4-2a)$$

$$= p_1(\beta; 0) p_1(\alpha; \rho_1); \quad \alpha, \beta > 0. \quad (C.4-2b)$$

That is, the normalized variables are independent. Note that the jamming conditions are present only in the SNR, ρ_1 .

C.5 CDF FOR SUM OF TWO NON-SIGNAL VARIABLES (PIC).

Since the two channels are independent, we may first derive the probability that the non-signal sum z_2 is greater than the signal channel sum z_1 , given a specific value of z_1 , then later average over z_1 to get the error probability. Formally,

$$\Pr\{z_2 > z_1 | z_1 = \alpha\} = 1 - F_2(\alpha), \quad (C.5-1)$$

J. S. LEE ASSOCIATES, INC.

where $F_2(\alpha)$ is the cumulative distribution function for z_2 . This function is found to be

$$\begin{aligned}
 F_2(\alpha) &= \Pr \left\{ z_2 = z_{21} + z_{22} \leq \alpha \right\} \\
 &= \int_0^\alpha \frac{dx}{(1+x)^2} \int_0^{\alpha-x} \frac{dy}{(1+y)^2} \\
 &= \int_0^\alpha \frac{dx}{(1+x)^2} \left[1 - \frac{1}{\alpha-x+1} \right] \\
 &= \frac{\alpha}{1+\alpha} - \int_0^\alpha \frac{dx}{(1+x)^2} \cdot \frac{1}{(\alpha-x+1)} . \tag{C.5-2}
 \end{aligned}$$

Using a partial-fraction expansion results in

$$\begin{aligned}
 \Pr \{ z_2 > z_1 | z_1 = \alpha \} &= \frac{1}{(\alpha+2)^2} \left\{ \int_0^\alpha dx \frac{x+3+\alpha}{(1+x)^2} + \int_0^\alpha \frac{dx}{1+\alpha-x} \right\} + \frac{1}{1+\alpha} \\
 &= \frac{2}{(\alpha+2)^2} [\alpha + 2 + \ln(\alpha+1)] . \tag{C.5-3}
 \end{aligned}$$

C.6 ERROR PROBABILITY FOR L=2 (PIC).

The pdf for z_1 is the convolution

$$\begin{aligned}
 p_{z1}(\alpha) &= p_1(\alpha; \rho_1) * p_1(\alpha; \rho_2) \\
 &= \int_0^\alpha dx p_1(x; \rho_1) p_1(\alpha-x; \rho_2) , \alpha > 0 . \tag{C.6-1}
 \end{aligned}$$

Thus the error probability is, conditioned on ρ_1 and ρ_2 ,

J. S. LEE ASSOCIATES, INC.

$$\begin{aligned}
 P_b(e|\rho_1, \rho_2) &= 2 \int_0^\infty d\alpha \frac{\alpha+2+\ln(\alpha+1)}{(\alpha+2)^2} \int_0^\alpha dx p_1(x;\rho_1) p_1(\alpha-x;\rho_2) \\
 &= 2 \int_0^\infty d\alpha \int_0^\infty dx p_1(\alpha;\rho_1) p_1(x;\rho_2) \cdot \frac{\alpha+x+2+\ln(\alpha+x+1)}{(\alpha+x+2)^2} \quad (C.6-2)
 \end{aligned}$$

Transforming the integration variables by

$$u = \frac{1}{1+\alpha}, \quad v = \frac{1}{1+x} \quad (C.6-3)$$

results in the expression

$$\begin{aligned}
 P_b(e|\rho_1, \rho_2) &= 2 \int_0^1 du \int_0^1 dv e^{-\rho_1 u - \rho_2 v} (1+\rho_1-\rho_1 u)(1+\rho_2-\rho_2 v) \\
 &\times \left(\frac{uv}{u+v} \right)^2 \left[\frac{u+v}{uv} + \ln \left(\frac{u+v}{uv} - 1 \right) \right]. \quad (C.6-4)
 \end{aligned}$$

Averaging over the jamming events (the number of hops jammed in the signal channel) yields the total error

$$\begin{aligned}
 P_b(e) &= (1-\gamma)^2 P_b(e|\rho_1=\rho_2=E_b/2N_0) \\
 &+ 2\gamma(1-\gamma) P_b(e|\rho_1=E_b/2N_0, \rho_2=E_b/2N_T) \\
 &+ \gamma^2 P_b(e|\rho_1=\rho_2=E_b/2N_T). \quad (C.6-5)
 \end{aligned}$$

J. S. LEE ASSOCIATES, INC.

APPENDIX D COMPUTER PROGRAM FOR SQUARE-LAW LINEAR COMBINING RECEIVER

The following pages contain the source code listing for the FORTRAN-77 program used for numerical computations of the performance of the square-law linear combining receiver for FH/RMFSK.

```

FDP-11 FORTRAN-77 V4.0-1 09:41:35 16-Jul-86 Page 1
RMFSKINHB.FTN;13 /F77/TR:BLOCKS/WR
0001 PROGRAM CRNHOP
C THIS PROGRAM COMPUTES THE ERROR PROBABILITY FOR RANDOM M-ARY
C FSK/FH WITH MULTIPLE HOPS/BIT IN THE PRESENCE OF PARTIAL-BAND
C NOISE JAMMING BY NUMERICAL INTEGRATION FOR THE SQUARE-LAW
C LINEAR COMBINING RECEIVER
C
C ANALYSIS: L. E. MILLER, R. H. FRENCH
C PROGRAM: R. H. FRENCH
C
C V 5.1.0 - COMPUTATIONS ONLY
C
C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
PARAMETER (L=125)
CHARACTER*13 FNAME, GNAME
LOGICAL TRASH1, TRASH2, HIGH, PRINTR
LOGICAL *1 GOOD
CHARACTER*1 REPLY, YES, NO
REAL*4 PRLOG(L), DBSR(L,J)
DIMENSION MATRIX(4,8), MLOW(4,8), MUPI(4,8), MIN(4,8), PIE(0.8)
VIRTUAL D(625), IDSUB(625), PRERR(625), IPSUB(625)
COMMON /RESET/ TRASH1, TRASH2
C COMMON /INPUTS/ PASSES PARAMETERS OF THE RUN
COMMON /INPUTS/ DEBNOL(5), LLIST(4), NSLOTS, GANLST(31), K, MM
C COMMON /SIZE/ PASSES NUMBERS OF PARAMETERS
COMMON /SIZE/ NO, NL, NG
COMMON /REGION/ HIGH
COMMON /PARMS/ BICK, LL
DATA YES, NO /'Y', 'N'/, PRINTR/ FALSE./
CALL ERSET(29, TRUE.., FALSE.., TRUE.., .15)
CALL GET(INJ, START, DBINC)
MORBIT=0.500*MM/MM-1.0D
DO 900 ILL=1, NL
LL=LLIST(IL)
FL=LL
DO 800 I0=1, MO
EBNO=10.0D0***(DEBNOL(10)/10.0D0)
RHOM=K*EBNO*FL
IOUT=DEBNOL(10)
DO 700 IG=1, NG
GAMMA=GANLST(IG)
MD=GAMMA*NSLOTS*D_500
C OPEN DATA FILE
C
C IOUT=GAMMA*1000. DO+0.500
WRITE(FNAME,730) MM, LL, IOUT, IOUT
FORMAT('P',11,11,12,2,14,4,'.DAT')
IF(PRINTR) THEN
OPEN(UNIT=6,FILE='FOR006.DAT', STATUS='OLD', FORM='FORMATTED')
ACCESS='APPEND'
730 CONTINUE
00013
00014
00015
00016
00017
00018
00019
00020
00021
00022
00023
00024
00025
00026
00027
00028
00029
00030
00031
00032
00033
00034
00035
00036
00037
00038
00039
00040
00041
00042
00043
00044
00045
00046
00047
00048
00049
00050
00051
00052
00053
00054
00055
00056
00057
00058
00059
00060
00061
00062
00063
00064
00065
00066
00067
00068
00069
00070
00071
00072
00073
00074
00075
00076
00077
00078
00079
00080
00081
00082
00083
00084
00085
00086
00087
00088
00089
00090
00091
00092
00093
00094
00095
00096
00097
00098
00099
00100
00101
00102
00103
00104
00105
00106
00107
00108
00109
00110
00111
00112
00113
00114
00115
00116
00117
00118
00119
00120
00121
00122
00123
00124
00125
00126
00127
00128
00129
00130
00131
00132
00133
00134
00135
00136
00137
00138
00139
00140
00141
00142
00143
00144
00145
00146
00147
00148
00149
00150
00151
00152
00153
00154
00155
00156
00157
00158
00159
00160
00161
00162
00163
00164
00165
00166
00167
00168
00169
00170
00171
00172
00173
00174
00175
00176
00177
00178
00179
00180
00181
00182
00183
00184
00185
00186
00187
00188
00189
00190
00191
00192
00193
00194
00195
00196
00197
00198
00199
00200
00201
00202
00203
00204
00205
00206
00207
00208
00209
00210
00211
00212
00213
00214
00215
00216
00217
00218
00219
00220
00221
00222
00223
00224
00225
00226
00227
00228
00229
00230
00231
00232
00233
00234
00235
00236
00237
00238
00239
00240
00241
00242
00243
00244
00245
00246
00247
00248
00249
00250
00251
00252
00253
00254
00255
00256
00257
00258
00259
00260
00261
00262
00263
00264
00265
00266
00267
00268
00269
00270
00271
00272
00273
00274
00275
00276
00277
00278
00279
00280
00281
00282
00283
00284
00285
00286
00287
00288
00289
00290
00291
00292
00293
00294
00295
00296
00297
00298
00299
00300
00301
00302
00303
00304
00305
00306
00307
00308
00309
00310
00311
00312
00313
00314
00315
00316
00317
00318
00319
00320
00321
00322
00323
00324
00325
00326
00327
00328
00329
00330
00331
00332
00333
00334
00335
00336
00337
00338
00339
00340
00341
00342
00343
00344
00345
00346
00347
00348
00349
00350
00351
00352
00353
00354
00355
00356
00357
00358
00359
00360
00361
00362
00363
00364
00365
00366
00367
00368
00369
00370
00371
00372
00373
00374
00375
00376
00377
00378
00379
00380
00381
00382
00383
00384
00385
00386
00387
00388
00389
00390
00391
00392
00393
00394
00395
00396
00397
00398
00399
00400
00401
00402
00403
00404
00405
00406
00407
00408
00409
00410
00411
00412
00413
00414
00415
00416
00417
00418
00419
00420
00421
00422
00423
00424
00425
00426
00427
00428
00429
00430
00431
00432
00433
00434
00435
00436
00437
00438
00439
00440
00441
00442
00443
00444
00445
00446
00447
00448
00449
00450
00451
00452
00453
00454
00455
00456
00457
00458
00459
00460
00461
00462
00463
00464
00465
00466
00467
00468
00469
00470
00471
00472
00473
00474
00475
00476
00477
00478
00479
00480
00481
00482
00483
00484
00485
00486
00487
00488
00489
00490
00491
00492
00493
00494
00495
00496
00497
00498
00499
00500
00501
00502
00503
00504
00505
00506
00507
00508
00509
00510
00511
00512
00513
00514
00515
00516
00517
00518
00519
00520
00521
00522
00523
00524
00525
00526
00527
00528
00529
00530
00531
00532
00533
00534
00535
00536
00537
00538
00539
00540
00541
00542
00543
00544
00545
00546
00547
00548
00549
00550
00551
00552
00553
00554
00555
00556
00557
00558
00559
00560
00561
00562
00563
00564
00565
00566
00567
00568
00569
00570
00571
00572
00573
00574
00575
00576
00577
00578
00579
00580
00581
00582
00583
00584
00585
00586
00587
00588
00589
00580
00591
00592
00593
00594
00595
00596
00597
00598
00599
00600
00601
00602
00603
00604
00605
00606
00607
00608
00609
00610
00611
00612
00613
00614
00615
00616
00617
00618
00619
00620
00621
00622
00623
00624
00625
00626
00627
00628
00629
00630
00631
00632
00633
00634
00635
00636
00637
00638
00639
00640
00641
00642
00643
00644
00645
00646
00647
00648
00649
00650
00651
00652
00653
00654
00655
00656
00657
00658
00659
00660
00661
00662
00663
00664
00665
00666
00667
00668
00669
00670
00671
00672
00673
00674
00675
00676
00677
00678
00679
00680
00681
00682
00683
00684
00685
00686
00687
00688
00689
00680
00691
00692
00693
00694
00695
00696
00697
00698
00699
00700
00701
00702
00703
00704
00705
00706
00707
00708
00709
00700
00711
00712
00713
00714
00715
00716
00717
00718
00719
00720
00721
00722
00723
00724
00725
00726
00727
00728
00729
00720
00731
00732
00733
00734
00735
00736
00737
00738
00739
00730
00741
00742
00743
00744
00745
00746
00747
00748
00749
00750
00751
00752
00753
00754
00755
00756
00757
00758
00759
00750
00761
00762
00763
00764
00765
00766
00767
00768
00769
00770
00771
00772
00773
00774
00775
00776
00777
00778
00779
00770
00781
00782
00783
00784
00785
00786
00787
00788
00789
00790
00791
00792
00793
00794
00795
00796
00797
00798
00799
00700
00801
00802
00803
00804
00805
00806
00807
00808
00809
00810
00811
00812
00813
00814
00815
00816
00817
00818
00819
00820
00821
00822
00823
00824
00825
00826
00827
00828
00829
00820
00831
00832
00833
00834
00835
00836
00837
00838
00839
00840
00841
00842
00843
00844
00845
00846
00847
00848
00849
00850
00851
00852
00853
00854
00855
00856
00857
00858
00859
00860
00861
00862
00863
00864
00865
00866
00867
00868
00869
00870
00871
00872
00873
00874
00875
00876
00877
00878
00879
00880
00881
00882
00883
00884
00885
00886
00887
00888
00889
00880
00891
00892
00893
00894
00895
00896
00897
00898
00899
00900
00901
00902
00903
00904
00905
00906
00907
00908
00909
00900
00911
00912
00913
00914
00915
00916
00917
00918
00919
00920
00921
00922
00923
00924
00925
00926
00927
00928
00929
00920
00931
00932
00933
00934
00935
00936
00937
00938
00939
00940
00941
00942
00943
00944
00945
00946
00947
00948
00949
00950
00951
00952
00953
00954
00955
00956
00957
00958
00959
00960
00961
00962
00963
00964
00965
00966
00967
00968
00969
00970
00971
00972
00973
00974
00975
00976
00977
00978
00979
00980
00981
00982
00983
00984
00985
00986
00987
00988
00989
00980
00991
00992
00993
00994
00995
00996
00997
00998
00999
00900
00911
00912
00913
00914
00915
00916
00917
00918
00919
00920
00921
00922
00923
00924
00925
00926
00927
00928
00929
00920
00931
00932
00933
00934
00935
00936
00937
00938
00939
00940
00941
00942
00943
00944
00945
00946
00947
00948
00949
00950
00951
00952
00953
00954
00955
00956
00957
00958
00959
00960
00961
00962
00963
00964
00965
00966
00967
00968
00969
00970
00971
00972
00973
00974
00975
00976
00977
00978
00979
00980
00981
00982
00983
00984
00985
00986
00987
00988
00989
00980
00991
00992
00993
00994
00995
00996
00997
00998
00999
00900
00911
00912
00913
00914
00915
00916
00917
00918
00919
00920
00921
00922
00923
00924
00925
00926
00927
00928
00929
00920
00931
00932
00933
00934
00935
00936
00937
00938
00939
00940
00941
00942
00943
00944
00945
00946
00947
00948
00949
00950
00951
00952
00953
00954
00955
00956
00957
00958
00959
00960
00961
00962
00963
00964
00965
00966
00967
00968
00969
00970
00971
00972
00973
00974
00975
00976
00977
00978
00979
00980
00981
00982
00983
00984
00985
00986
00987
00988
00989
00980
00991
00992
00993
00994
00995
00996
00997
00998
00999
00900
00911
00912
00913
00914
00915
00916
00917
00918
00919
00920
00921
00922
00923
00924
00925
00926
00927
00928
00929
00920
00931
00932
00933
00934
00935
00936
00937
00938
00939
00940
00941
00942
00943
00944
00945
00946
00947
00948
00949
00950
00951
00952
00953
00954
00955
00956
00957
00958
00959
00960
00961
00962
00963
00964
00965
00966
00967
00968
00969
00970
00971
00972
00973
00974
00975
00976
00977
00978
00979
00980
00981
00982
00983
00984
00985
00986
00987
00988
00989
00980
00991
00992
00993
00994
00995
00996
00997
00998
00999
00900
00911
00912
00913
00914
00915
00916
00917
00918
00919
00920
00921
00922
00923
00924
00925
00926
00927
00928
00929
00920
00931
00932
00933
00934
00935
00936
00937
00938
00939
00940
00941
00942
00943
00944
00945
00946
00947
00948
00949
00950
00951
00952
00953
00954
00955
00956
00957
00958
00959
00960
00961
00962
00963
00964
00965
00966
00967
00968
00969
00970
00971
00972
00973
00974
00975
00976
00977
00978
00979
00980
00981
00982
00983
00984
00985
00986
00987
00988
00989
00980
00991
00992
00993
00994
00995
00996
00997
00998
00999
00900
00911
00912
00913
00914
00915
00916
00917
00918
00919
00920
00921
00922
00923
00924
00925
00926
00927
00928
00929
00920
00931
00932
00933
00934
00935
00936
00937
00938
00939
00940
00941
00942
00943
00944
00945
00946
00947
00948
00949
00950
00951
00952
00953
00954
00955
00956
00957
00958
00959
00960
00961
00962
00963
00964
00965
00966
00967
00968
00969
00970
00971
00972
00973
00974
00975
00976
00977
00978
00979
00980
00981
00982
00983
00984
00985
00986
00987
00988
00989
00980
00991
00992
00993
00994
00995
00996
00997
00998
00999
00900
00911
00912
00913
00914
00915
00916
00917
00918
00919
00920
00921
00922
00923
00924
00925
00926
00927
00928
00929
00920
00931
00932
00933
00934
00935
00936
00937
00938
00939
00940
00941
00942
00943
00944
00945
00946
00947
00948
00949
00950
00951
00952
00953
00954
00955
00956
00957
00958
00959
00960
00961
00962
00963
00964
00965
00966
00967
00968
00969
00970
00971
00972
00973
00974
00975
00976
00977
00978
00979
00980
00981
00982
00983
00984
00985
00986
00987
00988
00989
00980
00991
00992
00993
00994
00995
00996
00997
00998
00999
00900
00911
00912
00913
00914
00915
00916
00917
00918
00919
00920
00921
00922
00923
00924
00925
00926
00927
00928
00929
00920
00931
00932
00933
00934
00935
00936
00937
00938
00939
00940
00941
00942
00943
00944
00945
00946
00947
00948
00949
00950
00951
00952
00953
00954
00955
00956
00957
00958
00959
00960
00961
00962
00963
00964
00965
00966
00967
00968
00969
00970
00971
00972
00973
00974
00975
00976
00977
00978
00979
00980
00981
00982
00983
00984
00985
00986
00987
00988
00989
00980
00991
00992
00993
00994
00995
00996
00997
00998
00999
00900
00911
00912
00913
00914
00915
00916
00917
00918
00919
00920
00921
00922
00923
00924
00925
00926
00927
00928
00929
00920
00931
00932
00933
00934
00935
00936
00937
00938
00939
00940
00941
00942
00943
00944
00945
00946
00947
00948
00949
00950
00951
00952
00953
00954
00955
00956
00957
00958
00959
00960
00961
00962
00963
00964
00965
00966
00967
00968
00969
00970
00971
00972
00973
00974
00975
00976
00977
00978
00979
00980
00981
00982
00983
00984
00985
00986
00987
00988
00989
00980
00991
00992
00993
00994
00995
00996
00997
00998
00999
00900
00911
00912
00913
00914
00915
00916
00917
00918
00919
00920
00921
00922
00923
00924
00925
00926
00927
00928
00929
00920
00931
00932
00933
00934
00935
00936
00937
00938
00939
00940
00941
00942
00943
00944
00945
00946
00947
00948
00949
00950
00951
00952
00953
00954
00955
00956
00957
00958
00959
00960
00961
00962
00963
00964
00965
00966
00967
00968
00969
00970
00971
00972
00973
00974
00975
00976
00977
00978
00979
00980
00981
00982
00983
00984
00985
00986
00987
00988
00989
00980
00991
00992
00993
00994
00995
00996
00997
00998
00999
00900
00911
00912
00913
00914
00915
00916
00917
00918
00919
00920
00921
00922
00923
00924
00925
00926
00927
00928
00929
00920
00931
00932
00933
00934
00935
00936
00937
00938
00939
00940
00941
00942
00943
00944
00945
00946
00947
00948
00949
00950
00951
00952
00953
00954
00955
00956
00957
00958
00959
00960
00961
00962
00963
00964
00965
00966
00967
00968
00969
00970
00971
00972
00973
00974
00975
00976
00977
00978
00979
00980
00981
00982
00983
00984
00985
00986
00987
00988
00989
00980
00991
00992
00993
00994
00995
00996
00997
00998
00999
00900
00911
00912
00913
00914
00915
00916
00917
00918
00919
00920
00921
00922
00923
00924
00925
00926
00927
00928
00929
00920
00931
00932
00933
00934
00935
00936
00937
00938
00939
00940
00941
00942
00943
00944
00945
00946
00947
00948
00949
00950
00951
00952
00953
00954
00955
00956
00957
00958
00959
00960
00961
00962
00963
00964
009
```


POP-11 FORTRAN-77 V4.0-1 09:41:56 16-Jul-86
 RMF SKINH8.FTN;13 /F77/TR:BLOCKS/MR

Page 5

POP-11 FORTRAN-77 V4.0-1 09:41:56 16-Jul-86
 RMF SKINH8.FTN;13 /F77/TR:BLOCKS/MR

Page 6

```

0001      C      SUBROUTINE GET(MJ,START,DBINC)
          C      INTERACTIVE INPUT OF PARAMETERS FOR RUN
          C
          C      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          C      CHARACTER*9 FIELD,BLANK9
          C      COMMON /INPUT/ DEBML(5),LLIST(4),MSLOTS,GAMLIST(31),K,MN
          C      C DEFAULT LISTS TEMPORARILY NEEDED ARE IN SHARED STORAGE WITH
          C      C THE LARGE CONVOLUTION WORKING ARRAYS
          C      COMMON /SHARE/ DG(31),DSNR(5,4)
          DATA BLANK9,' '
0002      32      WRITE(5,33)
0003      33      FORMAT(1X,BITS/SYMBOL_(K) [2]: '$')
0010      READ(5,3)K
0011      IF(K.EQ.0)K=2
0012      MN=2**K
0013      1      WRITE(5,2)
0014      2      FORMAT(1X, HOW MANY EB/NO? [1]: '$')
0015      READ(5,3)NO
0016      3      FORMAT(1Z)
0017      IF(NO.EQ.0)NO=1
0018      DO I=1,NO
0019      IF(K.LE.4) THEN
0020      DD=DSNR(IN,K)
0021      ELSE
0022      DD=0.00
0023      END IF
0024      4      WRITE(5,5)IN,DO
0025      5      FORMAT(1X,EB/NO(' ,12,' ) F9.6,1: '$')
0026      READ(5,6)FIELD
0027      6      FORMAT(A9)
0028      IF(FIELD.EQ.BLANK9) THEN
0029      DEBML(IN)=DO
0030      ELSE
0031      DECODE(9,61,FIELD)DEBML(IN)
0032      61     FORMAT(F9.6)
0033      END IF
0034      7      CONTINUE
0035      15     WRITE(5,16)
0036      16     FORMAT(1X, HOW MANY L? [4]: '$')
0037      READ(5,3)NL
0038      IF(NL.EQ.0)NL=4
0039      GO 21 IN=1,NL
0040      WRITE(5,19)IN,IN
0041      19      FORMAT(1X,L(' ,11,' ) F11.1,1: '$')
0042      READ(5,3)LIST(IN).EQ.0)LIST(IN)=IN
0043      IF(LLIST(IN).EQ.0)LLIST(IN)=IN
0044      21      CONTINUE
0045      22      WRITE(5,23)
0046      23      FORMAT(1X,HOPPING SLOTS? [2400]: '$')
0047      READ(5,24)NSLOTS

```

0001 SUBROUTINE PSUBE(RHOM, RHOT, LL, M, PE, D, IDSUB, NUSED, PRERR, IPSUB)

C COMPUTE UNCONDITIONAL ERROR PROBABILITY

C IMPLICIT DOUBLE PRECISION(A-H,O-Z)

C INTEGER JAM(8), LUP(8), JSUB(8)

C LOGICAL *1 GO, NONE, STORE

C LOGICAL TRASH1, TRASH2, HIGH

C WE DO WANT TO STORE ZERO ELEMENTS OF THE DENSITY FUNCTION.

C SINCE IT SAVES TIME TO AVOID REPEATING THE UNDERFLOWS

C VIRTUAL D(625), IDSUB(625)

C COMMON /REGION/ HIGH

C COMMON /RESET/ TRASH1, TRASH2

C COMMON /SHARE2/ LOW(8), LINC(8)

C DATA STORE/.TRUE./

C PE=0.00

C MPS=0

C JAM1=-1

C DO 199 11=0,LL

C JAM(1)=11

C ITER=1

C DO 101 I=ITER+1,M

C OUTERMOST NONSIGNAL LOOP ALWAYS STARTS FROM 0, BUT

C THE OTHERS START FROM THE CURRENT VALUE OF THE

C NEXT OUTER MORE LOOP TO PRODUCE THE SORTED EVENTS

C

C IF(I,L.GT.2) THEN

C JAM(1)=JAM(I-1)

C ELSE

C JAM(1)=0

C END IF

C 101 CONTINUE

C IF(JAM1.NE.JAM(1)) THEN

C UPDATE TEST VALUE FOR NEXT TIME, AND ...

C JAM(1)=JAM(1)

C TRASH1=.TRUE..

C END IF

C CALL EVENT(LL,M,JAM,PIE,D, IDSUB,NUSED)

C BYPASS CONDITIONAL ERROR PROBABILITY CALCULATION IF EVENT

C PROBABILITY IS ZERO. THIS SAVES MUCH TIME

C IF(PIE.EQ.0.0)GOTO 198

C STORE AS ELEMENT OF A SPARSE ARRAY TO SAVE MEMORY

C EVEN THOUGH WE STORE ZEROS. THE SORTING OF SUBSCRIPTS

C CUTS OUT MANY ELEMENTS.

C CALL LOCN(M,LOW,LUP,JAM,ISUB)

C CALL LOOKUP(PROB,PRERR,IPSUB,NPS,625,ISUB,STORE,NONE)

C IF IT IS NOT THERE, WE MUST COMPUTE IT

PDP-11 FORTRAN-77 V4.0-1 39:42:09 16-Jul-86 Page 9
 RMF SKINH8.FTN;13 /F77/TR-BLOCKS/MR

0001 C SUBROUTINE EVENT(LL,M,JMM,PIE,D,IDSUB,MUSED)
 C
 C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
 C LOGICAL*1 STORE,NONE
 C DIMENSION JMM(8),LUP(8)
 C VIRTUAL D(625),IDSUB(625)
 C COMMON /SHARE2/ LON(8),LMNC(8)
 C DATA STORE,/FALSE./
 C SET UP ARRAY DESCRIPTION DO:LL,...,0:LL) WITH M DIMENSIONS
 C DO I=1,M
 C LUP(I)=LL
 0002 1 CONTINUE
 C COMPUTE LINEAR EQUIVALENT SUBSCRIPT FOR JAMMING EVENT
 0003 CALL LOCN(M,1,0,LUP,JMM,TSUB)
 0004 C LOOK UP THE VALUE, GET 0,DO IF NOT THERE
 0005 CALL LOOKUP(PIE,D,IDSUB,MUSED,625,TSUB,STORE,NONE)
 0006 RETURN
 END

PDP-11 FORTRAN-77 V4.0-1 09:42:12 16-Jul-86 Page 10
 RMF SKINH8.FTN;13 /F77/TR-BLOCKS/MR

0001 SUBROUTINE GENPIE(LL,MM,MQ,NSLOTS,GOOD,MATRIX,MLOW,MINC,
 MUP,PIE,O,IDSUB,MUSED)
 C
 C SUBROUTINE TO GENERATE EVENT PROBABILITIES
 C
 C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
 C LOGICAL*1 GO,STORE,MONE,GOOD
 C DIMENSION MATRIX(LL,MM),MLOW(LL,MM),MINC(LL,MM),MUP(LL,MM),
 C PIE(O:MM),IWORK(8),LWORK(8),LUPMARK(8)
 C
 C VIRTUAL D(625),IDSUB(625)
 C STORE=.FALSE. => DON'T STORE ZERO ELEMENTS OF SPARSE ARRAY.
 C
 STORE=.FALSE.
 GOOD=.TRUE.
 IF(MQ.LE.0) THEN
 GOOD=.FALSE.
 RETURN
 END IF
 0002 9999
 0003 9999
 0004 9999
 0005 9999
 0006 9999
 0007 9999
 0008 9999
 0009 9999
 0010 9999
 0011 9999
 0012 9999
 0013 9999
 0014 9999
 0015 9999
 0016 9999
 0017 9999
 0018 9999
 0019 9999
 0020 9999
 0021 9999
 0022 9999
 0023 9999
 0024 9999
 0025 9999
 0026 9999
 0027 9999
 0028 9999
 0029 9999
 0030 9999
 0031 9999
 0032 9999
 0033 9999
 0034 9999
 0035 9999
 0036 9999
 0037 9999
 0038 9999
 0039 9999
 0040 9999
 0041 9999
 C
 C INITIALIZATION MATRIX-INITIAL LOOP
 C CALL MLINIT(MATRIX,MLOW,LL,MM)
 C
 C FORM COLUMN SUMS AND COMPUTE P(EVENT)
 P=PIE(K)
 101 CONTINUE
 C FORM JAMMING EVENT VECTOR
 DO 102 J=1,MM
 IWORK(J)=0
 DO 102 J=1,LL
 IWORK(J)=IWORK(J)+MATRIX(I,J)
 102 CONTINUE

```

PDP-11 FORTRAN-77 V4.0-1    09:42:12    16-Jul-86
F77/TR.BLOCKS/MR

RMF SKINHB.FTN.13

C SORT NONSIGNAL CHANNELS
      DO 103 1=2, NCH-1
      DO 103 J=1+1,NCH-1
      IF(IWORK(J) LT IWORK(1)) THEN
        ITEMP=IWORK(1)
        IWORK(1)=IWORK(J)
        IWORK(J)=ITEMP
      END IF
      CONTINUE
      CALL LOCN(NCH,LWORK,IWORK,ISUB)
      CALL LOOKUP(DOUT,D,1DSUB,1NUSED,625,1SUB,STORE,NONE)
      DOUT=DOUT+P
      CALL PUTIN(DOUT,D,1DSUB,1NUSED,625,1SUB,IERR,STORE)
      IF(IERR.NE.0) STOP 'TOO MANY EVENTS'
C ITERATE MATRIX-INDEX LOOP
      CALL MILIT(MATRIX,NLOW,NUP,MINC,LL,MP,GO)
      IF(GO) GOTO 999
      RETURN
END

```

PDP-11 FORTRAN-77 V4.0-1 09:02:18 16-Jul-86
 RMFSKINHB.FTN.1.3 /F77/TR:BLOCKS/MR

```

0021 30 DO 40 I=1,MUSE
0022 J=1
0023 IF (ICSUB(I) .EQ. K) GOTO 50
0024 CONTINUE
0025 RETURN

C REMOVE THE ZEROED ELEMENT AND BUMP COUNT OF ENTRIES USED
C
0026 50 DO 60 J=J,MUSE-1
0027   ICSub(1)=ICSub(1+1)
0028   C(1)=C(1+1)
0029   CONTINUE
0030   MUSE=MUSE-1
0031   RETURN
0032 END

```

Page 13
 PDP-11 FORTRAN-77 V4.0-1 09:42:21 16-Jul-86
 RMFSKINHB.FTN.1.3 /F77/TR:BLOCKS/MR
 Page 14

```

0001      SUBROUTINE LOOKUP(COUT,C,ICSUB,N,MMAX,K,STORE,NONE)
C THIS SUBROUTINE RETRIEVES AN ELEMENT OF A SPARSE ARRAY WHICH
C HAS BEEN STORED COMPACTLY BY STORING ONLY NON-ZERO ELEMENTS.
C THE ARRAY IS DOUBLE PRECISION.
C
C USAGE:
C   VIRTUAL ICSub(1:MAX), C(1:MAX)
C   LOGICAL *1 STORE, NONE
C   DOUBLE PRECISION COUT
C   CALL LOOKUP(COUT,C,ICSub,N,MMax,K,STORE,NONE)
C WHERE
C   COUT = VALUE OF C(K) (OUTPUT FROM SUBROUTINE)
C   C = ARRAY USED TO STORE NON-ZERO ELEMENTS
C   ICSub = AUXILIARY ARRAY TO STORE ACTUAL SUBSCRIPTS
C   N = NUMBER OF ELEMENTS OF C CURRENTLY IN USE
C   MAX = SIZE OF C
C   K = SUBSCRIPT OF SPARSE ARRAY TO LOOK UP
C   STORE = .TRUE. IF ZEROS STORED EXPLICITLY, ELSE .FALSE.
C   NONE = .FALSE. IF ZEROS NOT STORED OR ZEROS STORED AND
C                 ELEMENT IS FOUND IN THE STORED ARRAY
C                 .TRUE. IF ZEROS ARE STORED AND THE ELEMENT IS
C                 NOT FOUND (OUTPUT QUANTITY)
C
C PROGRAMMER: ROBERT H. FRENCH
C DATE: 11 JANUARY 1984
C
C
0002      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C   VIRTUAL ICSub(1:MAX), C(1:MAX)
C   LOGICAL *1 STORE, NONE
C   NONE=.FALSE.
0003   DO 10 I=1,N
0004     IF (ICSub(I) .NE. K) GOTO 10
0005     COUT=C(I)
0006   CONTINUE
0007   IF (ICSub(I) .NE. K) GOTO 10
0008   RETURN
0009   10 CONTINUE
0010   IF (STORE) THEN
0011     COUT=C(I)
0012     NONE=.TRUE..
0013   ELSE
0014     COUT=0.
0015   END IF
0016   RETURN
0017 END

```

```

0001      SUBROUTINE LOCN(NDIM,LL0W,LUP,ISUB,LINEAR)
C
C THIS SUBROUTINE COMPUTES THE EQUIVALENT LINEAR SUBSCRIPT FOR
C A MULTIDIMENSIONAL ARRAY OF NDIM DIMENSIONS
C
C IF THE ARRAY A IS DEFINED AS
C
C      DIMENSION A(IL0W(1):IUP(1),...,IL0W(NDIM):IUP(NDIM))
C AND ISUB(1),...,ISUB(NDIM) IS A SET OF SUBSCRIPTS FOR A,
C THEN THIS SUBROUTINE RETURNS IN LINEAR THE OFFSET FROM THE
C ORIGIN OF A TO THE ELEMENT A(ISUB(1),...,ISUB(NDIM)). ASSUMING
C THE FIRST SUBSCRIPT VARIES MOST RAPIDLY.
C
C USAGE:
C
C      DIMENSION IL0W(NDIM),IUP(NDIM),ISUB(NDIM)
C      DATA IL0W/ lower limits of defined subscripts of array/
C      DATA IUP/upper limits of defined subscripts of array/
C      ...SET ISUB TO DESIRED SEQUENCE OF SUBSCRIPTS...
C      CALL LOCN(NDIM,IL0W,IUP,ISUB,LINEAR)
C
C WHERE
C      NDIM = NUMBER OF DIMENSIONS THE ARRAY HAS
C      IL0W = ARRAY OF LOWER SUBSCRIPT BOUNDS
C      IUP = ARRAY OF UPPER SUBSCRIPT BOUNDS
C      ISUB = ARRAY CONTAINING SUBSCRIPT FOR WHICH LOCATION IS
C             TO BE COMPUTED
C      LINEAR = RETURNED VALUE OF OFFSET INTO ARRAY IN MEMORY
C
C
C PROGRAMMER: ROBERT H. FRENCH
C DATE: 11 JANUARY 1984
C
C      DIMENSION IL0W(NDIM),IUP(NDIM),ISUB(NDIM)
C      LINEAR=0
C      DO 10 I=1,NDIM-1
C      J=NDIM-I+1
C      LINEAR=(LINEAR+(ISUB(J)-IL0W(J)))*(IUP(J-1)-IL0W(J-1))+1
C      10 CONTINUE
C      LINEAR=LINEAR+ISUB(1)-IL0W(1)
C      RETURN
C
C
0002
0003
0004
0005
0006
0007
0008
0009
0010
C

```

```

0001      SUBROUTINE MLINIT(LMAT,LL0W,LMAXC,LMAXR)
C
C THIS SUBROUTINE INITIALIZES A "MATRIX DO-LOOP" STRUCTURE
C DEFINED BY THE FOLLOWING PSEUDO-FORTRAN CODE:
C
C      DO 100 LMAT(1,1)=LL0W(1,1),LUP(1,1),LINC(1,1)
C
C      :
C
C      DO 100 LMAT(1,2)=LL0W(1,2),LUP(1,2),LINC(1,2)
C
C      :
C
C      DO 100 LMAT(1,MAXC,2)=LL0W(1,MAXC,2),LUP(1,MAXC,2),LINC(1,MAXC,2)
C
C      :
C
C      DO 100 LMAT(1,MAXC,1)=LL0W(1,MAXC,1),LUP(1,MAXC,1),LINC(1,MAXC,1)
C
C      :
C
C      DO 100 LMAT(1,1)=LL0W(1,1),LUP(1,1),LINC(1,1)
C
C      :
C
C      (STATEMENTS IN RANGE OF LOOP)
C
C      100 CONTINUE
C
C      :
C
C      THE COMPANION ROUTINE MLITER HANDLES THE LOOP CONTROL AT THE
C      CONTINUE STATEMENT IN THE ABOVE STRUCTURE
C
C USAGE:
C
C      LOGICAL*1 GO
C      DIMENSION LMAT(1,MAXC,LMAXR),LL0W(1,MAXC,LMAXR),LUP(1,MAXC,LMAXR)
C      DIMENSION LINC(1,MAXC,LMAXR)
C
C      (INITIALIZE MATRIX LMAT TO STARTING VALUES OF THE NESTED LOOPS)
C      (INITIALIZE MATRIX LL0W TO STOPPING VALUES OF THE NESTED LOOPS)
C      (INITIALIZE MATRIX LUP TO INCREMENTS OF THE LOOPS)
C      (INITIALIZE MATRIX LINC TO INCREMENTS OF THE LOOPS)
C      CALL MLINIT(LMAT,LL0W,LMAXC,LMAXR)
C
C      100 CONTINUE
C
C      :
C      (STATEMENTS IN RANGE OF LOOPS)
C
C      CALL MLITER(LMAT,LL0W,LUP,LINC,LMAXC,LMAXR,GO)
C
C WHERE
C
C      LMAT = ARRAY FOR STORAGE OF LOOP INDICES. LMAT(1,1) IS THE
C             OUTER-MOST LOOP; LMAT(1,MAXC,LMAXR), THE INNER-MOST LOOP.
C
C      LL0W = ARRAY FOR STORAGE OF LOOP STARTING VALUES. IN SAME
C             SEQUENCE AS LMAT
C
C      LUP = ARRAY FOR STORAGE OF LOOP FINAL VALUES. IN SAME
C             SEQUENCE AS LMAT
C
C      LINC = ARRAY FOR STORAGE OF LOOP INCREMENTS. IN SAME
C             SEQUENCE AS LMAT
C
C      LMAY = NUMBER OF LOOPS NESTED
C      GO = LOGICAL VARIABLE. .TRUE. IF JUMP BACK TO BEGINNING OF
C             STATEMENTS IN THE RANGE OF THE LOOP SHOULD OCCUR.
C             .FALSE. OTHERWISE I.E. OUTER-MOST LOOP TERMINATED)
C
C

```

PDP-11 FORTRAN-77 V4.0-1 09-42-.6 16-JU1-86 Page 17
 RMFSKNIHB.FTN;13 /F77/TR:BLOCKS/MR

```

C PROGRAMMER: ROBERT H. FRENCH          DATE: 10 MARCH 1986
C
 0002 DIMENSION LMAT(LMAXC,LMAXR),LLON(LMAXC,LMAXR)
 0003 DO 1 M=1,LMAXR
 0004 DO 1 N=1,LMAXC
 0005 LMAT(N,N)=LLON(M,N)
 0006 1 CONTINUE
 0007 RETURN
 0008 END

```

PDP-11 FORTRAN-77 V4.0-1 09-42-.6 16-JU1-86 Page 18
 RMFSKNIHB.FTN;13 /F77/TR:BLOCKS/MR

```

0001      SUBROUTINE MLITER(LMAT,LLON,LUP,LINC,LMAXC,LMAXR,GO)
C
C LOOP ITERATION LOGIC FOR A "MATRIX DO-LOOP"
C
C SEE DETAILED COMMENTS IN SUBROUTINE MLINIT FOR USAGE AND
C PARAMETER DEFINITIONS
C
C PROGRAMMER: ROBERT H. FRENCH
C
C DATE: 10 MARCH 1986
C
C
 0002 LOGICAL*1 GO
 0003 DIMENSION LMAT(LMAXC,LMAXR),LLON(LMAXC,LMAXR),LUP(LMAXC,LMAXR)
 0004 DIMENSION LINC(LMAXC,LMAXR)
 0005 GO=. TRUE.
 0006 DO 100 MDX=1,LMAXR
 0007 NSUB=LMAXR+1-MDX
 0008 DO 100 MDX=1,LMAXC
 0009 MSUB=LMAXC+1-MDX
 0010 LMAT(1,MSUB)=LMAT(1,MSUB)+LINC(1,MSUB,MSUB)
 0011 IF((LINC(1,MSUB,MSUB).GE.0.AND.LMAT(1,MSUB).LE.LUP(1,MSUB,MSUB))
     OR.
     $ (LINC(1,MSUB,MSUB).LT.0.AND.LMAT(1,MSUB,MSUB).GE.LUP(1,MSUB,MSUB)))
 0012 RETURN
 0013 LMAT(1,MSUB,MSUB)=LLON(1,MSUB,MSUB)
 0014 CONTINUE
 0015 GO=. FALSE.
 0016 RETURN
END

```

PDP-11 FORTRAN-77 V4.0-1
RMF SKN1H8.FTN;13 /F77/TR:BLOCKS/WR

09:42:31 16-Jul-86

Page 19 PDP-11 FORTRAN-77 V4.0-1 /F77/TR:BLOCKS/WR
RMF SKN1H8.FTN;13 /F77/TR:BLOCKS/WR

0001 SUBROUTINE PR1HOP(KJAM,KM,KO,XN,AIN)

```
C THIS SUBROUTINE COMPUTES THE EVENT PROBABILITY FOR ALL
C POSSIBLE JAMMING PATTERNS WITH NON-ZERO PROBABILITY FOR
C L=1 HOP/SYMBOL FOR RMF SKNFH IN PBNJ
C
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C AIM=0.0D
C ! IF THIS IS AN IMPOSSIBLE CASE, RETURN WITH RESULT = 0.0
C IF(KJAM.GT.MINO(KQ,KM)) RETURN
C KMAX=KJAM-1
C LMAX=KM-KJAM-1
C JMAX=KA-1
C IMAX=MAX0(KPMAX,LPMAX,JPMAX)
C PROC=.1.D0
C Q=KQ
C DIFFNQ=KN-KQ
C EN=KN
C DO 100 LOOP=0,IMAX
C F=LOOP
C IF(LOOP.LE.KPMAX) PROD=PROD*(Q-F)
C IF(LOOP.LE.JPMAX) PROD=PROD/(EN-F)
C IF(LOOP.LE.LPMAX) PROD=PROD*(DIFFNQ-F)
C 100 CONTINUE
C AIM=PROD
C RETURN
C END
```

```
0001 C INITIALIZATION OF BLOCK DATA
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C COMMON /SHARE/ D6(31) DSNR(5,4)
C COMMON /SHARE2/ LOW(8),LINC(8)
C DEFAULT LISTS FOR INTERACTIVE PARAMETER INPUTS
C ARE SHARED WITH LARGE WORKING STORAGE ARRAYS SINCE THEY
C MAY BE DESTROYED ONCE THE INPUT PARAMETERS ARE SET UP
C DATA D6 / .001D0, .002D0, .005D0,
C           .01D0, .02D0, .05D0,
C           .1D0, .2D0, .5D0, 1.0D0, 21*0.0D0/
C DATA DSNR /13,3524700, 12,313300, 10,9444300, 0.00, 0.00,
C           10,60657200, 9,629400, 8,3524800, 0.00, 0.00,
C           9,0940100, 8,169000, 6,97199500, 0.00, 0.00,
C           8,0763500, 7,199600, 6,06964600, 0.00, 0.00/
C FREQUENTLY NEEDED CONSTANT ARRAYS AND SCALARS
C DATA LOW(8*D),LINC(8*D)/
C END
```

```

0001      SUBROUTINE PSEL1(JSUB,LL,MM,RHON,RHOT,PROB)
          C RANDOM MFSK/FPP IN PARTIAL BAND TONE JAMMING,
          C GIVEN A JAMMING EVENT
          C JSUB - JAMMING EVENT VECTOR
          C LL - NUMBER OF HOPS/SYMBOL
          C MM - ALPHABET SIZE
          C PROB - RESULTING CONDITIONAL ERROR PROBABILITY
          C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          C EXTERNAL PGRAND,DGAU20
          C DIMENSION WORK(75), STACK(75), SAVE(75)
          C INTEGER JSUB(8)
          C INTEGER NCHAN(0:6)
          C COMMON /PARMS/ BIGK, LL
          C AJ = NONCENTRAL PARAMETER FOR JAMMED HOPS
          C ANJ = NONCENTRAL PARAMETER FOR NON-JAMMED HOPS
          C NJ = NUMBER OF JAMMED HOPS
          C MNJ = NUMBER OF NON-JAMMED HOPS
          C COMMON /PUPAR/ AJ, ANJ, NJ, MNJ
          C COMMON /JAMCNT/ NCHAN
          C LL=LL
          C BIGK=RHON/RHOT
          C NJ=JSUB(1)
          C MNJ=LL-NJ
          C AJ=2.0D+NMJ*RHOT
          C ANJ=2.0D+MNJ*RHON
          C ASHIFT=DMAX1(AJ,ANJ)
          C SUBTRACT 1 FOR BENEFIT OF RAPID CHI-SQUARE DENSITY CALCULATION
          C NJ=AJ-1
          C MNJ=MNJ-1
          C COUNT NUMBER OF NON-SIGNAL CHANNELS WITH L HOPS JAMMED
          C
          DO 1 I=0,LL
          NCHAN(1)=0
          1  CONTINUE
          DO 2 I=2,MM
          JSUB=JSUB(1)
          NCHAN(JSUB)=NCHAN(JSUB)+1
          2  CONTINUE
          CALL DG15(PGRAND,ASHIFT,TAIL)
          CALL ADQUAD(0,ASHIFT,BODY,DGAU20,PGRAND,1.0-10,
                      WORK,STACK,75,KODE)
          3  IF(KODE.NE.0) THEN
              WRITE(5,3) KODE
              3  FORMAT(5,3) KODE
              STOP 'FATAL ERROR'
          END IF
          30  PROB=TAIL+BODY
          RETURN
          0034      END

```

```

0001      DOUBLE PRECISION FUNCTION PGRAND(BETA)
          C INTEGRAND FUNCTION
          C
          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          LOGICAL TRASH2
          VIRTUAL PUV(1:2053), IPUT(1:2053), IFLT(1:8191),
          $ VIRTUAL FLV(8191), FLT(8191), IFLT(8191)
          0002      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          0003      COMMON /PARMS/ BIGK, LL
          0004      $ VIRTUAL PUV(1:2053), IPUT(1:2053), IFLT(1:8191),
          0005      $ INTEGER NCHAN(0:6),
          0006      COMMON /JAMCNT/ NCHAN
          0007      COMMON /RESET/ TRASH1, TRASH2
          0008      COMMON /PARMS/ BIGK, LL
          0009      C AJ = NONCENTRAL PARAMETER FOR JAMMED HOPS
          0010      C ANJ = NONCENTRAL PARAMETER FOR NON-JAMMED HOPS
          0011      C NJ = NUMBER OF JAMMED HOPS
          0012      C MNJ = NUMBER OF NON-JAMMED HOPS
          0013      C COMMON /PUPAR/ AJ, ANJ, NJ, MNJ
          0014      C COMMON /JAMCNT/ NCHAN
          0015      C LL=LL
          0016      C BIGK=RHON/RHOT
          0017      C NJ=JSUB(1)
          0018      C MNJ=LL-NJ
          0019      C AJ=2.0D+NMJ*RHOT
          0020      C ANJ=2.0D+MNJ*RHON
          0021      C ASHIFT=DMAX1(AJ,ANJ)
          0022      C SUBTRACT 1 FOR BENEFIT OF RAPID CHI-SQUARE DENSITY CALCULATION
          0023      C NJ=AJ-1
          0024      C MNJ=MNJ-1
          0025      C COUNT NUMBER OF NON-SIGNAL CHANNELS WITH L HOPS JAMMED
          0026      C
          0027      PROB=1.00
          0028      DO 10 I=0,LL
          0029      IF(NCHAN(I).NE.0) THEN
          0030          ISUB=ISUB(1)
          0031          JSUB=JSUB
          0032          I=IFLT(ISUB)
          0033          IT=IFLT(ISUB)
          0034          IF(IT.EQ.0,DO,AND, IT.EQ.0) THEN
          0035              NOT FOUND. COMPUTE IT AND ENTER INTO TABLE
          0036              X=F(BETA,I)
          0037              FLV(I)=X
          0038              FLT(ISUB)=BETA
          0039              I=IFLT(ISUB)+1
          END IF
          20      C
          0040      END

```

```

Page 23
PDP-11 FORTRAN-77 V4.0-1 09:42:38 16-Jul-86
RMSKINH8.FIN.i3 /F77/TR/BLOCKS/WR

0040      ELSE IF (IT.NE.BETA .OR. IT.EQ.I+1) THEN
          NOT THIS ENTRY, TRY NEXT ONE
          ISUB=ISUB+1
          IF (ISUB.GT.8191) ISUB=ISUB-8191
          IF (ISUB.NE.JSUB) THEN
              GOTO 20
          ELSE
              HASH TABLE OVERFLOW, MUST COMPUTE, CAN NOT STORE
              X=FL(BETA,I)
          END IF
          ELSE IF (IT.EQ.BETA .AND. IT.EQ.I+1) THEN
              G01 ITI
              K=FLV(ISUB)
          END IF
          PROD=PROD*OX((X,NCHAN(I)))
          END IF
          CONTINUE
          ISUB=IHASH(BETA,2053)
0045      JSUB=ISUB
          T=PUI((ISUB))
          30     I=IPUT((ISUB))
          0046      IF (I.EQ.0.00 .AND. IT.EQ.0) THEN
              NOT FOUND, COMPUTE IT AND ENTER INTO TABLE
              Y=PUI((ISUB))=Y
              PUI((ISUB))=BETA
              IPUT((ISUB))=NJI+2
              ELSE IF (IT.NE.BETA .OR. IT.EQ.NJ+2) THEN
                  NOT THIS ENTRY, TRY NEXT
                  ISUB=ISUB+1
                  IF (ISUB.GT.2053) ISUB=ISUB-2053
                  IF (ISUB.NE.JSUB) THEN
                      GOTO 30
                  ELSE
                      HASH TABLE OVERFLOWED
                      Y=PUI((ISUB))
                  END IF
                  ELSE IF (IT.EQ.BETA .AND. IT.EQ.NJ+2) THEN
                      G01 ITI!
                      Y=PUI((ISUB))
                  END IF
                  PGRAD=Y*(1.00-PROD)
              RETURN
          END
0047      0048      C
          0049      0050      0051      0052      0053      0054      0055      0056      0057      0058      0059      0060      0061      0062      0063      0064      0065      0066      0067      0068      0069      0070      0071      0072      0073      0074      0075      0076

```

```

POP-11 FORTRAN-77 V4.0-1   09:42:46  16-Jul-56
      /F77/TR-BLOCKS/WR
      RMFSKINH8.FTN;13

0001      DOUBLE PRECISION FUNCTION PUI(Y)
0002      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003      DIMENSION WORK(75), STACK(75), SAVE(75)
0004      EXTERNAL DGENV1, PUIG
0005      COMMON /PULCOM/ Y
0006      C          AJ = NONCENTRAL PARAMETER FOR JAMMED HOPS
0007      C          ANJ = NONCENTRAL PARAMETER FOR NON-JAMMED HOPS
0008      C          NJ = NUMBER OF JAMMED HOPS
0009      C          MNJ = NUMBER OF NON-JAMMED HOPS
0010      C          COMMON /PUIPAR/ AJ, ANJ, NJ, MNJ
0011      C          COMMON /PARMS/ BIGK, LL
0012      C          IF(Y.LE.0.D0) THEN
0013          PUI=0.D0
0014          RETURN
0015      END IF
0016      C          IF(NJ.NE.-1.AND. NJ.NE.LL-1) THEN
0017          C          WE MUST CONVOLVE TWO NONCENTRAL CHI-SQUARE DENSITIES
0018          YY=Y
0019          CALL ADQUA2(0.,Y, VALUE, DGENV1, PUIG, 1.D-10, WORK, STACK,
0020              SAVE, 75, KODE)
0021          C          IF(KODE.NE.0) THEN
0022              1          WRITE(5,1) KODE
0023                  1          FORMAT(1,1) KODE
0024          END IF
0025          C          CALL CHISQE(Y,LL,-1,ANJ,F,KODE)
0026          C          ALL HOPS UNJAMMED
0027          C          CALL CHISQE(Y,LL,-1,ANJ,F,KODE)
0028          C          ELSE IF(NJ.EQ.LL-1) THEN
0029              C          ALL HOPS JAMMED
0030              C          CALL CHISQE(Y,BIGK,LL,-1,AJ,F,KODE)
0031              C          F=F/BIGK
0032          END IF
0033          C          IF(KODE.NE.0) THEN
0034              C          WRITE(5,111) KODE
0035                  1          FORMAT(1,1) KODE
0036          END IF
0037          C          STOP 'FATAL ERROR'
0038      END IF

```

```

PDP-11 FORTRAN-77 V4.0-1 09:42:49 16-Jul-86
RMFS:NMH.FTM:13 /F77/TR:BLOCKS/MR
0001      C DOUBLE PRECISION FUNCTION PU16(X)
          C INNER INTEGRAND FUNCTION
          C
          0002      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          COMMON /PU16C/ Y
          AJ = NONCENTRAL PARAMETER FOR JAMMED HOPS
          AMJ = NONCENTRAL PARAMETER FOR NON-JAMMED HOPS
          NJ = NUMBER OF JAMMED HOPS
          MNJ = NUMBER OF NON-JAMMED HOPS
          COMMON /PU16PAR/ AJ, AMJ, NJ, MNJ
          COMMON /PARMS/ BIGK, LL
          CALL CHISQE(X,BIGK, NJ, AJ, F1, KODE)
          IF(KODE.NE.0) THEN
          WRITE(5,1) KODE
          FORMAT(1, BESEL FUNCTION ERROR CODE: ,12)
          STOP 'FATAL IN JAMMED HOP DENSITY'.
          END IF
          CALL CHISQE(YY-X, MNJ, AJ, F2, KODE)
          IF(KODE.NE.0) THEN
          WRITE(5,1) KODE
          STOP 'FATAL IN UNJAMMED HOP DENSITY'.
          END IF
          PU16=F1+F2
          RETURN
        END
  1
  0013
  0014
  0015
  0016
  0017
  0018
  0019

```

Page 25

```

PDP-11 FORTRAN-77 V4.0-1 09:42:51 16-Jul-86
RMFS:NMH.FTM:13 /F77/TR:BLOCKS/MR
0001      C DOUBLE PRECISION FUNCTION FL(ALPHA,L)
          C
          C NONSIGNAL CHANNEL CUMULATIVE DISTRIBUTION
          C
          0002      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          COMMON /PARMS/ BIGK, LL
          IF(L.NE.0.AND.L.NE.-LL) THEN
          R=(BIGK-L)/BIGK
          BIGLOG=LOG(BIGK)
          ARG=L*BIGLOG-ALPHA/2.
          START=DEXP(-ARG)
          FL1=L-1
          DO 100 M=0,150
          EN=N
          IF(N.EQ.0) THEN
          PART=1.00
          TERM=DEXP(ALPHA/2.00,START,LL-1)
          SUM=TERM
          ELSE
          PART=PART*R*(EN+FL1)/EN
          TERM=PART*DEXP(ALPHA/2.00,START,LL-1)
          DUMMY=SUM+TERM
          SUM=DUMMY
          END IF
          IF(DABS(SUM)).LE.1.0-11*DABS(SUM)) GOTO 125
          CONTINUE
          STOP 'FL SUM DID NOT CONVERGE'
          ELSE
          IF(L.EQ.0) THEN
          A=ALPHA/2.00
          ELSE IF(L.EQ. LL) THEN
          A=ALPHA/(2.00*BIGK)
          END IF
          START=DEXP(-A)
          SUM=DEXP(A,START,LL-1)
          END IF
          FL=L-1.00-SUM
          RETURN
        END

```

Page 26

PDP-11 FORTRAN-77 V4.0-1 09:42:55 16-Jul-85 Page 27

RMF SKINH8.FTN;13 /F77/TR:BLOCKS/WR

```

0001      C DOUBLE PRECISION FUNCTION DEXP(X,START,IUP)
          C INCOMPLETE EXPONENTIAL FUNCTION (DOUBLE PRECISION)
          C
          C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          TERM=START
          DEXP=TERM
          IF(IUP.EQ.0) RETURN
          DO 100 I=1,IUP
          F=1
          TERM=TERM*X/F
          DEXP=DEXP+TERM
          CONTINUE
          RETURN
        END
100
0010
0011
0012

```

PDP-11 FORTRAN-77 V4.0-1 09:42:56 16-Jul-86 Page 28

RMF SKINH8.FTN;13 /F77/TR:BLOCKS/WR

```

0001      C SUBROUTINE CHISQE(X,N,A,DEN,KODE)
          C NON-CENTRAL CHI-SQUARE DENSITY FOR EVEN DEGREES OF FREEDOM
          C DEGREES OF FREEDOM (M) IS M=2*N+2
          C
          C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          B=DSQRT(X*A)
          CALL DXBESI(B,N,BESSEL,KODE)
          IF(KODE.NE.0) RETURN
          R=X/A
          IF(R.NE.0.D0) THEN
            POWER=R**N/(N/2.D0)
          ELSE
            IF(N.NE.0) THEN
              POWER=0.D0
            ELSE
              POWER=1.D0
            END IF
          END IF
          DEN=0.5D0*POWER*DEXP(B-0.5D0*(X+A))*BESSEL
        RETURN
      END

```

```

PDP-11 FORTRAN-77 V4.0-1 09:42:58 16-JU
RMFSKIN1B.FTN;13 /F77/TR-BLOCKS/MR

0001      SUBROUTINE ADQUA2(XL,XU,Y,OR,F,T)
          C
          C ADAPTIVE QUADRATURE ALGORITHM
          C XL - LOWER LIMIT OF INTEGRAL (IN)
          C XU - UPPER LIMIT OF INTEGRAL (IN)
          C Y - VALUE OF INTEGRAL (OUT)
          C QR - NAME OF A QUADRATURE RULE SUBROUTINE
          C WITH CALLING SEQUENCE
          C     CALL QR(XL,XU,F,Y)
          C
          C F - NAME OF FUNCTION TO BE INTEGRATE
          C TOL - ERROR TOLERANCE FOR FINAL ANSWER
          C WORK - WORK ARRAY OF SIZE N (IN)
          C STACK- SECOND WORK ARRAY OF SIZE N, MUST
          C        SAME ARRAY AS WORK (IN)
          C SAVE- THIRD WORK ARRAY OF SIZE N, MUST
          C        SAME ARRAY AS WORK NOR SAME AS S
          C N - SIZE OF WORK AND STACK; MAX. NO.
          C KODE - ERROR INDICATOR (OUT)
          C       0 -- NO ERROR
          C       1 -- WORK ARRAYS TOO SMALL
          C       2 -- EPS DIVIDED TO ZERO, EITHER
          C            TIGHT A TOLERANCE OR ROUND
          C            ATTAINING REQUIRED ACCURACY
          C
          C R. H. FRENCH, 14 AUGUST 1984

          C
          C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          C EXTERNAL F
          C DIMENSION WORK(N),STACK(N),SAVE(N)
          C KODE=0
          C Y=0.0D0
          C WORK(1)=XU
          C A=XL
          C MPTS=1
          C EPS=TOL
          C STACK(1)=EPS
          C CALL QR(XL,XU,F,T)
          C SAVE(1)=T
          C B=WORK(MPTS)
          C 10   XM=(A+B)*0.5D0
          C     CALL OR(A,XM,F,P1)
          C     CALL OR(XM,B,F,P2)
          C     IF(DABS(T-P2).LE.EPS) GOTO 20
          C     SPLIT IT
          C     MPTS=MPTS+1
          C     IF(MPTS.GT.N) THEN
          C       KODE=1
          C     RETURN
          C   END IF
          C   WORK(MPTS)=XM
          C   EPS=EPS/2.0D0
          C
          C
          0002 0003 0004 0005 0006 0007 0008 0009 0010 0011 0012 0013 0014 0015 0016 0017 0018 0019 0020 0021 0022 0023 0024 0025

```

Page: 29
 09:42:58 16-Jul-86
 /F77/TR:BLOCKSIWR
 3 77 V4.0-1

SUBROUTINE AQUA2(XL,XU,Y,OR,F,TOL,WORK,STACK,SAVE,N,KODE)

LIVE QUADRATURE ALGORITHM

- LOWER LIMIT OF INTEGRAL (IN)
- UPPER LIMIT OF INTEGRAL (IN)
- VALUE OF INTEGRAL (OUT)
- NAME OF A QUADRATURE RULE SUBROUTINE (IN)
- WITH CALLING SEQUENCE
CALL OR(XL,XU,F,Y)
- NAME OF FUNCTION TO BE INTEGRATED (IN)
- ERROR TOLERANCE FOR FINAL ANSWER (IN)
- WORK ARRAY OF SIZE N (IN)
- SECOND WORK ARRAY OF SIZE N, MUST NOT BE SAME ARRAY AS WORK (IN)
- THIRD WORK ARRAY OF SIZE N, MUST NOT BE SAME ARRAY AS WORK NOR SAME AS STACK (IN)
- SIZE OF WORK AND STACK; MAX. NO. OF BISECTIONS (IN)
- ERROR INDICATOR (OUT)

```

PDP-11 FORTRAN-77 V4.0-1 09:42:58 16-Jul-86 Page 30
RMFSKNIHR.FTN;13 /F77/TR:BLOCKS/MR

0026 IF(EPS.EQ.0.00) THEN
0027   KODE=2
0028   RETURN
0029 END IF
0030 STACK(NPTS)=EPS
0031 T=P1
0032 SAVE(NPTS)=P2
0033 GOTO 10
C FINISHED A PIECE
20 Y=Y+P1+P2
EPS=STACK(NPTS)
T=SAVE(NPTS)
NPTS=NPTS-1
A=B
IF(NPTS.EQ.0) RETURN
GOTO 10
END

```

PDP-11 FORTRAN-77 V4.0-1 09:43:01 16-Jul-86 Page 31

RMF SKINH8.F1N,13 /F77/TR:BLOCKS/MR

0001 FUNCTION IHASH(BETA,ISIZE)

0002 C AD HOC HASHING FUNCTION

0003 C IMPLICIT DOUBLE PRECISION(A-H,O-Z)

0004 IF(BETA.LT.1.00) THEN

0005 IF(BETA.GT.0.500) THEN

0006 B=1.00/(1.01D0-BETA)

0007 ELSE

0008 B=50.00+1.500/(BETA+0.01100)

0009 END IF

0010 ELSE

0011 B=100000.00*(BETA-DINT(BETA*1000.00)/1000.00)+1B-7.D0

0012 END IF

0013 B=B*23.00

0014 SIZE=ISIZE

0015 I=DM001(R+0.500,SIZE)*0.500

0016 IHASH=I+1

0017 RETURN

PDP-11 FORTRAN-77 V4.0-1 09:43:03 16-Jul-86 Page 32

RMF SKINH8.FTN,13 /F77/TR:BLOCKS/MR

0001 SUBROUTINE DG16(A,B,F,ANSWER)

0002 C 16-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL

0003 C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4

0004 C R. H. FRENCH, 28 FEBRUARY 1986

0005 C

0006 IMPLICIT DOUBLE PRECISION (A-H,O-Z)

0007 DIMENSION X(8),W(8)

0008 DATA X/ 0.09501250983763744018500,

0009 S/ 0.281603555077925891323000,

0010 S/ 0.45801677765722738634200,

0011 S/ 0.61787624440264374844700,

0012 S/ 0.75540440835500303389500,

0013 S/ 0.86563120238783174388000,

0014 S/ 0.94457502307323257697800,

0015 S/ 0.98940093499154993299600 /

0016 DATA W/ 0.18945061045506849628500,

0017 S/ 0.18260341504492358886700,

0018 S/ 0.16915651939500253818900,

0019 S/ 0.149595988881657673208100,

0020 S/ 0.1246289712555387205200,

0021 S/ 0.09515851168249278481000,

0022 S/ 0.0622535293854789286300,

0023 S/ 0.02715245941175409485200 /

0024 ANSWER=0.00

0025 BMA02=(B-A)/2.00

0026 BPA02=(B+A)/2.00

0027 DO 10 I=1,8

0028 C=X(I)*BMA02

0029 0010 Y1=BPA02-C

0030 0011 Y2=BPA02-C

0031 0012 ANSWER=ANSWER+W(I)*(F(Y1)+F(Y2))

0032 10 CONTINUE

0033 ANSWER=ANSWER+BMA02

0034 RETURN

0035 END

PDP-11 FORTRAN-77 V4.0-1 09:43:06 16-Jul-86

RMF SKINH8.FTN:13 /F77/TR:BLOCKS/WR

Page 33

PDP-11 FORTRAN-77 V4.0-1 09:43:08 16-Jul-86

RMF SKINH8.FTN:13 /F77/TR:BLOCKS/WR

Page 34

0001 SUBROUTINE DGAU7(A,B,F,ANSWER)

C C 16-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL.

C C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4

C C R. H. FRENCH, 28 FEBRUARY 1986

C C IMPLICIT DOUBLE PRECISION (A-H,O-Z)

C C DIMENSION X(8),W(8)

C C DATA X/ 0.0950125098376374401850D0,

C C 0.281603550779228913230D0,

C C 0.4580167776572273863420D0,

C C 0.6178762444026437484770D0,

C C 0.755044083550030338950D0,

C C 0.8656312023878317436890D0,

C C 0.945750240732325760780D0,

C C 0.9894009349916499329960D0,

C C 1.08940610455684962850D0,

C C 0.1826034150449235888670D0,

C C 0.1691565193950025381199D0,

C C 0.14998598881657673208100D0,

C C 0.124628971255338720520D0,

C C 0.0951585116824927848100D0,

C C 0.0522535239386478928630D0,

C C 0.0271524594117540948520D0,

C C ANSWER=0.00

C C BPA02=(B-A)/2.00

C C BPA02=(B-A)/2.00

C C DO 10 I=1,8

C C C=X(1)*BPA02

C C Y1=BPA02+C

C C Y2=BPA02-C

C C ANSWER=ANSWER+W(I)*(F(Y1)+F(Y2))

C C CONTINUE

C C ANSWER=ANSWER*BPA02

C C RETURN

C C END

10

D-18

```

PDP-11 FORTRAN-77 V4.0-1    09:43:08    16-Jul-86
RMF SKINH8.FTN:13    /F77/TR:BLOCKS/WR
0001    SUBROUTINE DLAG15(F,A,RESULT)
C    C DOUBLE PRECISION 15-POINT LAGUERRE INTEGRATION
C    C FOR THE SHIFTED SEMI-INFINITE INTERVAL:
C    C
C    C    INFINITY
C    C    /
C    C    F(X) DX = RESULT
C    C    /
C    C    A
C    C
C    C    IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C    C    DIMENSION X(15),W(15)
C    C    DATA X/ 0.09330781201700,
C    C       0.0942669174030200,
C    C       0.09515641207100,
C    C       0.09604952620400,
C    C       0.096944952620400,
C    C       0.097833662741400,
C    C       0.098722856801900,
C    C       0.09961622661300,
C    C       0.1005051916975400,
C    C       0.1013928248217600,
C    C       0.1022664402215100,
C    C       0.10318040779833000,
C    C       0.104056440779833000,
C    C       0.104925788944900,
C    C       0.10577477538400,
C    C       0.10662272175100,
C    C       0.10746633662741400,
C    C       0.10830644900,
C    C       0.1091622856801900,
C    C       0.110022856801900,
C    C       0.11087876321200,
C    C       0.11173277100500,
C    C       0.112622856801900,
C    C       0.1135088557268660,
C    C       0.1143944877538400,
C    C       0.11527772100500,
C    C       0.11615020566400,
C    C       0.11700177844300,
C    C       0.117876321200,
C    C       0.118753117142300,
C    C       0.119647519765300,
C    C       0.12053117142300,
C    C       0.1214544877538400,
C    C       0.122366440221500,
C    C       0.12325664333464000,
C    C       0.124106333464000,
C    C       0.125006333464000,
C    C       0.12589477538400,
C    C       0.1267740419268300,
C    C       0.127664333464000,
C    C       0.12853117142300,
C    C       0.12942544877538400,
C    C       0.130317876321200,
C    C       0.13121544877538400,
C    C       0.13211435020566400,
C    C       0.133013277100500,
C    C       0.1339122856801900,
C    C       0.134811277100500,
C    C       0.135710268660,
C    C       0.13660925788944900,
C    C       0.137508248217600,
C    C       0.1383972377100500,
C    C       0.13928622661300,
C    C       0.1401751916975400,
C    C       0.141064088557268660,
C    C       0.14195398881657673208100,
C    C       0.1428438772553387205200,
C    C       0.14373376647892863000,
C    C       0.144623655824927848100,
C    C       0.145513523938647892863000,
C    C       0.14640341504492358886700,
C    C       0.147293304492358886700,
C    C       0.14818320341504492358886700,
C    C       0.149073102387831743689000,
C    C       0.1500630013257607800,
C    C       0.15105289712553387205200,
C    C       0.1520427865824927848100,
C    C       0.153032675824927848100,
C    C       0.15402256516824927848100,
C    C       0.155012454492358886700,
C    C       0.156002343492358886700,
C    C       0.157002232492358886700,
C    C       0.158002121492358886700,
C    C       0.159002010492358886700,
C    C       0.160001900492358886700,
C    C       0.1610017900492358886700,
C    C       0.1620016800492358886700,
C    C       0.1630015700492358886700,
C    C       0.1640014600492358886700,
C    C       0.1650013500492358886700,
C    C       0.1660012400492358886700,
C    C       0.1670011300492358886700,
C    C       0.1680010200492358886700,
C    C       0.1690009100492358886700,
C    C       0.1700008000492358886700,
C    C       0.1710006900492358886700,
C    C       0.1720005800492358886700,
C    C       0.1730004700492358886700,
C    C       0.1740003600492358886700,
C    C       0.1750002500492358886700,
C    C       0.1760001400492358886700,
C    C       0.1770000300492358886700,
C    C       0.1780009200492358886700,
C    C       0.1790008100492358886700,
C    C       0.1800007000492358886700,
C    C       0.1810005900492358886700,
C    C       0.1820004800492358886700,
C    C       0.1830003700492358886700,
C    C       0.1840002600492358886700,
C    C       0.1850001500492358886700,
C    C       0.1860000400492358886700,
C    C       0.1870009300492358886700,
C    C       0.1880008200492358886700,
C    C       0.1890007100492358886700,
C    C       0.1900006000492358886700,
C    C       0.1910004900492358886700,
C    C       0.1920003800492358886700,
C    C       0.1930002700492358886700,
C    C       0.1940001600492358886700,
C    C       0.1950000500492358886700,
C    C       0.1960009400492358886700,
C    C       0.1970008300492358886700,
C    C       0.1980007200492358886700,
C    C       0.1990006100492358886700,
C    C       0.2000005000492358886700,
C    C       0.2010003900492358886700,
C    C       0.2020002800492358886700,
C    C       0.2030001700492358886700,
C    C       0.2040000600492358886700,
C    C       0.2050009500492358886700,
C    C       0.2060008400492358886700,
C    C       0.2070007300492358886700,
C    C       0.2080006200492358886700,
C    C       0.2090005100492358886700,
C    C       0.2100004000492358886700,
C    C       0.2110002900492358886700,
C    C       0.2120001800492358886700,
C    C       0.2130000700492358886700,
C    C       0.2140009600492358886700,
C    C       0.2150008500492358886700,
C    C       0.2160007400492358886700,
C    C       0.2170006300492358886700,
C    C       0.2180005200492358886700,
C    C       0.2190004100492358886700,
C    C       0.2200003000492358886700,
C    C       0.2210001900492358886700,
C    C       0.2220000800492358886700,
C    C       0.2230009700492358886700,
C    C       0.2240008600492358886700,
C    C       0.2250007500492358886700,
C    C       0.2260006400492358886700,
C    C       0.2270005300492358886700,
C    C       0.2280004200492358886700,
C    C       0.2290003100492358886700,
C    C       0.2300002000492358886700,
C    C       0.2310000900492358886700,
C    C       0.2320009800492358886700,
C    C       0.2330008700492358886700,
C    C       0.2340007600492358886700,
C    C       0.2350006500492358886700,
C    C       0.2360005400492358886700,
C    C       0.2370004300492358886700,
C    C       0.2380003200492358886700,
C    C       0.2390002100492358886700,
C    C       0.2400001000492358886700,
C    C       0.2410000900492358886700,
C    C       0.2420009800492358886700,
C    C       0.2430008700492358886700,
C    C       0.2440007600492358886700,
C    C       0.2450006500492358886700,
C    C       0.2460005400492358886700,
C    C       0.2470004300492358886700,
C    C       0.2480003200492358886700,
C    C       0.2490002100492358886700,
C    C       0.2500001000492358886700,
C    C       0.2510000900492358886700,
C    C       0.2520009800492358886700,
C    C       0.2530008700492358886700,
C    C       0.2540007600492358886700,
C    C       0.2550006500492358886700,
C    C       0.2560005400492358886700,
C    C       0.2570004300492358886700,
C    C       0.2580003200492358886700,
C    C       0.2590002100492358886700,
C    C       0.2600001000492358886700,
C    C       0.2610000900492358886700,
C    C       0.2620009800492358886700,
C    C       0.2630008700492358886700,
C    C       0.2640007600492358886700,
C    C       0.2650006500492358886700,
C    C       0.2660005400492358886700,
C    C       0.2670004300492358886700,
C    C       0.2680003200492358886700,
C    C       0.2690002100492358886700,
C    C       0.2700001000492358886700,
C    C       0.2710000900492358886700,
C    C       0.2720009800492358886700,
C    C       0.2730008700492358886700,
C    C       0.2740007600492358886700,
C    C       0.2750006500492358886700,
C    C       0.2760005400492358886700,
C    C       0.2770004300492358886700,
C    C       0.2780003200492358886700,
C    C       0.2790002100492358886700,
C    C       0.2800001000492358886700,
C    C       0.2810000900492358886700,
C    C       0.2820009800492358886700,
C    C       0.2830008700492358886700,
C    C       0.2840007600492358886700,
C    C       0.2850006500492358886700,
C    C       0.2860005400492358886700,
C    C       0.2870004300492358886700,
C    C       0.2880003200492358886700,
C    C       0.2890002100492358886700,
C    C       0.2900001000492358886700,
C    C       0.2910000900492358886700,
C    C       0.2920009800492358886700,
C    C       0.2930008700492358886700,
C    C       0.2940007600492358886700,
C    C       0.2950006500492358886700,
C    C       0.2960005400492358886700,
C    C       0.2970004300492358886700,
C    C       0.2980003200492358886700,
C    C       0.2990002100492358886700,
C    C       0.3000001000492358886700,
C    C       0.3010000900492358886700,
C    C       0.3020009800492358886700,
C    C       0.3030008700492358886700,
C    C       0.3040007600492358886700,
C    C       0.3050006500492358886700,
C    C       0.3060005400492358886700,
C    C       0.3070004300492358886700,
C    C       0.3080003200492358886700,
C    C       0.3090002100492358886700,
C    C       0.3100001000492358886700,
C    C       0.3110000900492358886700,
C    C       0.3120009800492358886700,
C    C       0.3130008700492358886700,
C    C       0.3140007600492358886700,
C    C       0.3150006500492358886700,
C    C       0.3160005400492358886700,
C    C       0.3170004300492358886700,
C    C       0.3180003200492358886700,
C    C       0.3190002100492358886700,
C    C       0.3200001000492358886700,
C    C       0.3210000900492358886700,
C    C       0.3220009800492358886700,
C    C       0.3230008700492358886700,
C    C       0.3240007600492358886700,
C    C       0.3250006500492358886700,
C    C       0.3260005400492358886700,
C    C       0.3270004300492358886700,
C    C       0.3280003200492358886700,
C    C       0.3290002100492358886700,
C    C       0.3300001000492358886700,
C    C       0.3310000900492358886700,
C    C       0.3320009800492358886700,
C    C       0.3330008700492358886700,
C    C       0.3340007600492358886700,
C    C       0.3350006500492358886700,
C    C       0.3360005400492358886700,
C    C       0.3370004300492358886700,
C    C       0.3380003200492358886700,
C    C       0.3390002100492358886700,
C    C       0.3400001000492358886700,
C    C       0.3410000900492358886700,
C    C       0.3420009800492358886700,
C    C       0.3430008700492358886700,
C    C       0.3440007600492358886700,
C    C       0.3450006500492358886700,
C    C       0.3460005400492358886700,
C    C       0.3470004300492358886700,
C    C       0.3480003200492358886700,
C    C       0.3490002100492358886700,
C    C       0.3500001000492358886700,
C    C       0.3510000900492358886700,
C    C       0.3520009800492358886700,
C    C       0.3530008700492358886700,
C    C       0.3540007600492358886700,
C    C       0.3550006500492358886700,
C    C       0.3560005400492358886700,
C    C       0.3570004300492358886700,
C    C       0.3580003200492358886700,
C    C       0.3590002100492358886700,
C    C      
```

POP-11 FORTRAN-77 V4.0-1
RMFSKINH8.FTN;13

Page 35

POP-11 FORTRAN-77 V4.0-1
RMFSKINH8.FTN;13

16-Jul-86

Page 36

SUBROUTINE PSEL2(JSUB,LL,MM,RHON1,RHOT1,PROB)

C RANDOM MFSK/FH IN PARTIAL BAND TONE JAMMING,
C GIVEN A JAMMING EVENT

C JSUB - JAMMING EVENT VECTOR
C LL - NUMBER OF HOPS/SYMBOL
C MM - ALPHABET SIZE
C PROB - RESULTING CONDITIONAL ERROR PROBABILITY

C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
INTEGER RO,RL
INTEGER JSUB(8)

DIMENSION M(0:3),CO(0:24),CL(0:24),
COEFL(10),COEFK(10),CEFP(10),
INDEX(20),INDUP(20)

\$ \$
C SYMBOLIC NAMES FOR INDEX-IDENTIFICATION SUBSCRIPTS
C (LOOP NUMBER(MODULE=1),
PARAMETER(LOOPR=1,LOOPK=2,LOOPP=0),
COMMON /PARMS/ BIGK, LL
COMMON /RRRCOM/ RHOT, RHON

C COMPUTE PARAMETERS
C RHON=RHON1
C RHOT=RHOT1
C LL=LL1
C BIGK=RHON/RHOT
C LM1=LL-1
C FK11 = DXI(BIGK-1,DO,LM1)

C COMPUTE POWERS OF FL
C DO 10 I=0,LL
C N(I)=0

0015 10 CONTINUE
0016 DO 12 I=2,MM
0017 12 CONTINUE
0018 N(JSUB(I))=N(JSUB(I))+1

C OVER-ALL SUMMATION INITIALIZATION
C SUM=0.00

C --- START LOOP ON RO ---
C DO 9000 RO=0,N(0)

0022 FRO=R0
0023 IF (PO.EQ.0) THEN
0024 COFO=1.00
0025 ELSE
0026 COFO=-COEOF*((N(0)-FRO+1.00)/FRO)

0027 END IF

0028

D-19

0029 C --- START LOOP ON KO ---
C KOMAX=RO*LM1

C PRE-COMPUTE THE J.C.P. MILLER COEFFICIENTS OVER KO!
CALL JCMPF(CO,KOMAX,RO,LM1)
DO 8000 KO=0,KOMAX

C --- START LOOP ON RL ---
C DO 7000 RL=0,M(LL)
C FRL=RL
IF (RL.EQ.0) THEN
COEFL=1.00
ELSE
COEFL=COEFL*((M(LL)-FRL+1.00)/FRL)

0030 END IF

C --- START LOOP ON KL ---
C KLMAX=RL*LM1

C PRE-COMPUTE THE J.C.P. MILLER COEFFICIENTS OVER KL!
CALL JCMPF(CL,KLMAX,RL,LM1)
DO 6000 KL=0,KLMAX

IF (KL.EQ.0) THEN
POWKL=1.00
ELSE
POWKL=POWKL/BIGK

END IF
COEFL=CL(KL)*POWKL

C --- START THE VARIABLE-LEVEL NESTED LOOPS
C LOOPS=3*LM1
C IF (LOOPS.EQ.0) GOTO 555

C SET UP INDEX OF OUTERMOST LOOP
C INDEX(1)=0
C SET UP THE NON-VARYING UPPER LIMITS
C DO 22 I=1,LM1
C INDUP(3*I-2)=N(1)

0031 22 CONTINUE

C MARK OUTER-MOST LOOP AS JUST ITERATED

C NSUB=1

C PERFORM INITIALIZATION CODE FOR LOOPS FROM JUST-ITERATED LOOP
C INWARD

Page 37
C
POP-11 FORTRAN-77 V4.0-1 09:43:11 16-Jul-86
RMFSK\118.FTM;13 /F77/TR:BLOCKS/MR

PDP-11 FORTRAN-77 V4.0-1 09:43:24 16-Jul-86 Page 39
 RMFSKIN18.FTM;13 /F77/TR:BLOCKS/WR
 C DOUBLE PRECISION FUNCTION DEF(LL,NEST,IR,K,IP,BIGK)
 C THE FUNCTION 0(P) FOR BMJ/RMFSK
 C
 0002 IMPLICIT DOUBLE PRECISION(A-H,O-Z)
 0003 DIMENSION G(0:8)
 0004 IF(LL.EQ.2) THEN
 0005 IF(NEST.EQ.1) THEN
 0006 DEF=DX(BIGK,TR-K)
 0007 IF(MOD(K,2).EQ.1) DEF=-DEF
 0008 END IF
 0009 ELSE IF(LL.EQ.3) THEN
 0010 IF(NEST.EQ.1) THEN
 0011 DEF=DX((2.00*BIGK,IR-K)*DBINCO(K,IP))*
 0012 (IF(MOD(K,2).EQ.1) DEF=-DEF
 0013 ELSE IF(NEST.EQ.2) THEN
 0014 DEF=DBINCOL(IR-K,IP)*DX((BIGK*(BIGK-2.00),IR-K-1)*
 0015 DX((BIGK-1.00,IP))
 0016 END IF
 0017 ELSE IF(LL.EQ.4) THEN
 0018 IF(NEST.EQ.1) THEN
 0019 DENOM=3.00*BIGK*BIGK-3.00*BIGK+1.00
 0020 COEF1=(2.00*BIGK*BIGK-3.00*BIGK+1.00)/DENOM
 0021 G(1)=K*COEF1
 0022 BK1=8*BIGK,1.00
 0023 COEF2=0.500*BK1*BK1/DENOM
 0024 DO Q1=N=2,IP
 0025 G(N)=(K+1-N)*COEF1*G(N-1) + (2*K+2-N)*COEF2*G(N-2)/N
 0026 CONTINUE
 0027 DEF=DX((BIGK,3*(IR-K))*DX((DENOM,K)*G(IP))
 0028 IF(MOD(K,2).EQ.1) DEF=-DEF
 0029 ELSE IF(NEST.EQ.2) THEN
 0030 TUP=MIND(IP,K)
 0031 ILGW=MAX0(0,IP-IR+K)
 0032 BK3=8*BIGK-3.00
 0033 TR1=3.00*BIGK-1.00
 0034 BASE=8K3*BIGK/TR1
 0035 SUMQ=0.00
 0036 DO 42 IQ=ILGW,IUP
 0037 SUMQ=SUMQ+DBINCOL(IR-K,IP-1)*DBINCO(K,10)*
 0038 *DX((BK3,IR-K-IP)*DX((TR1,K)*DX((BASE,1Q))
 0039 CONTINUE
 0040 DEF=SUMQ/DX((BIGK,IR+IR-K-K-IP)*DX((BIGK-1,IP))
 0041 G(0)=1.00
 0042 DENOM=DX((BIGK,3)-3.00*BIGK*BIGK+3.00*BIGK
 0043 COEF1=(BIGK*BIGK-3.00*BIGK+2.00)/DENOM
 0044 G(1)=(IR-K)*COEF1
 0045 BK=BIGK-1.00
 0046 COEF2=0.500*BK1*BK1/(BIGK+DENOM)

POP-11 FORTRAN-77 V4.0-1 09:43:24 16-Jul-86 Page 40
 RMFSKIN18.FTM;13 /F77/TR:BLOCKS/WR
 DO 43 N=2,IP
 G(N)=(IR-K+1-N)*COEF1*G(N-1)+
 (2*(IR-K+1)-N)*COEF2*G(N-2))/N
 0047 CONTINUE
 0048 DEE=DX((DENOM,IR-K)*G(IP))
 0049 IF(MOD(K,2).EQ.1) DEE=-DEF
 0050 END IF
 0051 RETURN
 0052 END IF
 0053 END IF
 0054 RETURN
 0055 END

```

POP-11 FORTRAN-77 V4.0-1   09:43:33    16-Jul-86
RMFSKINH8.FTM;13 /F77/TR:BLOCKS/MR
0001      C FUNCTION ICAPP(LL,NEST,IR,K)
          C COMPUTE UPPER SUMMATION LIMIT PL
          C
          C IF(LL.LE.2) THEN
          0002        ICAPP=0
          0003        ELSE IF(LL.EQ.3) THEN
          0004          IF(NEST.EQ.1) THEN
          0005            ICAPP=K
          0006            ELSE IF(NEST.EQ.2) THEN
          0007              ICAPP=TR-K
          0008              END IF
          0009            ELSE IF(LL.EQ.4) THEN
          0010              IF(NEST.EQ.1) THEN
          0011                ICAPP=2^K
          0012                ELSE IF(NEST.EQ.2) THEN
          0013                  ICAPP=IR
          0014                  ELSE IF(NEST.EQ.3) THEN
          0015                    ICAPP=2*(IR-K)
          0016                    END IF
          0017                    END IF
          0018                    RETURN
          0019
          0020

```

```

Page 41
POP-11 FORTRAN-77 V4.0-1   09:43:35    16-Jul-86
RMFSKINH8.FTM;13 /F77/TR:BLOCKS/MR
0001      C DOUBLE PRECISION FUNCTION GRAL(A0,IB0,LL)
          C THE INNER INTEGRAL FOR CONDITIONAL ERROR PROBABILITY
          C
          C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
          0002        COMMON /RRCOM/ RHOT,RHOM
          0003        COMMON /PAMPS/ BIGK,LL
          0004        A0=1.0D+AO
          0005        IF(LL.GT.0 .AND. LL.LT.LL) THEN
          0006          BK1=BIGK-1.D0
          0007          BASE=BK1/(1.D0+BIGK*A0)
          0008          ARG1=(LL-LL)*RHOM/A1
          0009          AR2=L1*RHOT*BIGK/(1.D0+BIGK*A0)
          0010          ARG2=AR2*A1/BK1
          0011          POWER=1.D0
          0012          SUM=0.0
          0013          DO 100 I0=0,IB0
          0014            IF(IQ.GT.0) POWER=POWER*BASE
          0015            PM=DQRT(POWER)
          0016            SUM=GMLGPM(IQ*LL-1,IB0-IQ,ARG1,PM)+SUM
          0017            CONTINUE
          0018            100
          0019            F=1.D0
          0020            DO 110 I1=1,IB0
          0021              F=F*(I1/A1)
          0022            CONTINUE
          110          F=(FDIX(A1,LL-L1))/DXI(1.0D+BIGK*A0,LL)
          0023          F=(FDIX(A1,LL-L1))/DXI(1.0D+BIGK*A0,LL)
          0024          GRAL=SUM*(DEP/(ARG1+ARG2)*A0)
          0025          ELSE IF(LL.EQ.0) THEN
          0026            ARG1=-LL*RHOM/A1
          0027            START=DEP(ARG1*A0-LL*DLOG(A1))
          0028            DO 200 I=1,IB0
          0029              START=START*(I/A1)
          0030            CONTINUE
          0031            GRAL=GMLGPM(LL-1,IB0,ARG1,START)
          0032            ELSE IF(LL.EQ.LL) THEN
          0033              A2=1.D0+BIGK*A0
          0034              ARG1=-LL*RHOT/A2
          0035              BASE=BIGK/A2
          0036              START=DEP(ARG1*BIGK*A0-LL*DLOG(A2))
          0037              DO 300 I=1,IB0
          0038                START=START*(I*BASE)
          0039                CONTINUE
          0040                GRAL=GMLGPM(LL-1,IB0,ARG1,START)
          0041            END IF
          0042            RETURN
          0043

```

PPB-11 FORTRAN-77 V4.0-1 09:43:40 16-Jul-86 Page 43
RMF SKINH8.FTM;13 /F77/TR:BLOCKS/MR

```
0001      DOUBLE PRECISION FUNCTION GNLGPM( IALFA, N, X, PREMUL )
C
C GENERALIZED LAGUERRE POLYNOMIALS WITH PREMULTIPLICATION BY
C A WEIGHTING FACTOR (TO DEFER THE INEVITABLE OVERFLOWS)
C
0002      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003      TERM=OBINCOIN+(IALFA,N)*PREMUL
0004      SUM=TERM
0005      IF(N.EQ.0) GOTO 200
0006      DO 100 M=1,N
0007      TERM=TERM*((X/M)*((N-M+1.D0)/(IALFA+M)))
0008      SUM=SUM+TERM
0009      100   CONTINUE
0010      200   GNLGPM=SUM
0011      RETURN
0012      END
```

PPB-11 FORTRAN-77 V4.0-1 09:43:42 16-Jul-86 Page 44
RMF SKINH8.FTM;13 /F77/TR:BLOCKS/MR

```
0001      SUBROUTINE JCPMF( C, KMAX, R, LM1 )
C
C COMPUTE J.C.P. MILLER COEFFICIENTS DIVIDED BY K!
C
0002      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003      INTEGER R
0004      DIMENSION C(0:KMAX)
0005      C(0)=1.0D0
0006      IF(LM1.EQ.0) RETURN
0007      DO 100 K=1,KMAX
0008      SUM=0.0D0
0009      MUP=MINO(K,LM1)
0010      BC=1.0D0
0011      DO 90 N=1,MUP
0012      BC=BC*((K-N+1.D0)/N)
0013      SUM=SUM+BC*((R+1.D0)**N-K)*C(K-N)
0014      90   CONTINUE
0015      C(K)=SUM/K
0016      100   CONTINUE
0017      FAC=1.D0
0018      DO 110 K=1,KMAX
0019      FAC=FAC*K
0020      C(K)=C(K)/FAC
0021      110   CONTINUE
0022      RETURN
0023      END
```

J. S. LEE ASSOCIATES, INC.

APPENDIX E COMPUTER PROGRAM FOR INDIVIDUAL CHANNEL ADAPTIVE GAIN CONTROL RECEIVER

The following pages contain the source code listing for the FORTRAN-77 program used for numerical computations of the performance of the individual channel adaptive gain control receiver for FH/RMFSK.

PREVIOUS PAGE
IS BLANK

PDP-11 FORTRAN-77 V4.0-1 08:32:51 16-Jul-86 Page 1
 ICAGC.FTN;22 /F77/TR-BLOCKS/MR

```

PROGRAM AGCIC
C THIS PROGRAM COMPUTES THE ERROR PROBABILITY FOR RANDOM M-ARY
C FSK/FH WITH MULTIPLE HOPS/BIT IN THE PRESENCE OF PARTIAL-BAND
C NOISE JAMMING WITH INDIVIDUAL CHANNEL AGC RECEIVER
C BY NUMERICAL INTEGRATION
C
C ANALYSIS: L. E. MILLER, R. H. FRENCH
C PROGRAM: R. H. FRENCH
C V 1.0.0 - COMPUTATIONS ONLY
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      PARAMETER (IJ=126)
      CHARACTER*13 FNAME, GRNAME
      CHARACTER*1 REPLY, YES, NO
      REAL*4 PRLOG(IJ), DEBNOL(IJ)
      COMMON /INPUTS/ PASSES PARAMETERS OF THE RUN
      COMMON /INPUTS/ DEBNOL(5), ILLIST(4), NSLOTS, GAMLIST(31), K, NM
      COMMON /SIZE/ PASSES NUMBERS OF PARAMETERS
      COMMON /SIZE/ NO, NL, NG
      DATA YES, NO /'Y', 'N'/
      CALL GET(NJ, START, 081NC)
      CALL ERSETT29(.TRUE., .FALSE., .TRUE., .FALSE., .15)
      NORBIT=0.500*PI/(NM-1.00)
      DO 900 NL=1,NL
      LI=ILLIST(IJ)
      FLL=LL
      DO 800 10=1,NL
      EBNO=10.0D+0*(DEBNOL(10)/10.0D)
      RHON=K*EBNO/FLL
      IJOUT=DEBNOL(10)
      DO 700 IG=1,NL
      GAMMA=GAMLIST(IG)
      NM=NM+NSLOTS+0.500
      C OPEN DATA FILE
      C
      IJOUT=GMAPP*1000.0D+0.5D0
      WRITE(FNAME,730) NM, LL, IJOUT, IJOUT
      FORMAT('1', I1, I1, I2, 2, 14, 4, 'DAT')
      WRITE(6,776) NM, LL, DEBNOL(10), GAMMA
      FORMAT('1', M=, I2, 5X, 'L= ', I2, 5X, 'GAMMA= ', I1P10.3/
      S, SRR(0B), IJX, PIE)
      WRITE(5,733) FNAME
      FORMAT(4, FILE=NAME, STATUS='OLD', FORM='UNFORMATTED',
      S, ERR=750)
      OPEN(UNIT=4, FILE=NAME, STATUS='OLD', FORM='UNFORMATTED',
      S, ERR=750)
      C HAVE AN EXISTING FILE, READ TO SEE HOW FAR WE GOT BEFORE
      C
  
```

PDP-11 FORTRAN-77 V4.0-1 08:32:51 16-Jul-86 Page 2
 ICAGC.FTN;22 /F77/TR-BLOCKS/MR

```

      0031          WRITE(5,1100)
      0032          1100  FORMAT('1', IS THIS A FIX-UP RE-RUN? [Y/N]: ', $)
      0033          READ(5,1101) REPLY
      0034          1101  FORMAT(A1)
      0035          IF(REPLY.EQ.'NO') GOTO 1300
      C WE ARE TO FIX UP AN EXISTING FILE WHICH HAS GARBAGE AT THE HIGH END
      C
      C ... FIRST READ EVERYTHING
      C ... READ(4) MMIN, LLIN, EBNOIN, NSLIN, GAMIN, DBSR, PRLOG
      C       IF(MMIN.NE.MM .OR. LLIN.NE.LL .OR. EBNOIN.NE. DEBNOL(10)
      C       .OR. GAMIN.NE. GAMMA .OR. NSLIN.NE. NSLOTS)
      C       STOP 'FILE SYNC ERROR OR CORRUPTED FILE'
      C
      0036          WRITE(5,1102)
      0037          1102  FORMAT('1', HOW MANY POINTS ARE GOOD? ', $)
      0038          READ(5,1103) JGOOD
      0039          1103  FORMAT(I3)
      0040          0040
      0041          0041
      0042          0042
      0043          0043
      C CREATE A NEW FILE WITH THE GOOD PART OF THE DATA IN IT
      C IN PROGRESS FORMAT
      OPEN(UNIT=4, FILE=NAME, STATUS='NEW', FORM='UNFORMATTED')
      0044          0044
      0045          WRITE(4) NM, LL, DEBNOL(10), NSLOTS, GAMMA
      0046          DO 1120 16000=1,36000
      0047          WRITE(4) DBSR(16000), PRLOG(16000)
      0048          1120  CONTINUE
      0049          CLOSE(UNIT=4)
      C AND GO FINISH UP THE CALCULATIONS
      0050          0050
      C NOT A FIX-UP RUN, MUST BE A CONTINUATION RUN
      C
      0051          1300  READ(4) MMIN, LLIN, EBNOIN, NSLIN, GAMIN
      0052          1300  IF(MMIN.NE.MM .OR. LLIN.NE.LL .OR. EBNOIN.NE. DEBNOL(10)
      C       .OR. GAMIN.NE. GAMMA .OR. NSLIN.NE. NSLOTS)
      C       STOP 'FILE SYNC ERROR OR CORRUPTED FILE'
      JJ=0
      0053          0053
      0054          740   JJ=JJ+1
      0055          READ(4, END=742) DBSR(JJ), PRLOG(JJ)
      0056          GOTO 740
      0057          742   CLOSE(UNIT=4)
      0058          0058
      C NO EXISTING FILE, THIS IS THE FIRST TIME: CREATE FILE HEADER RECORD
      0059          750   JJ=1
      0060          OPEN(UNIT=4, FILE=NAME, STATUS='NEW', FORM='UNFORMATTED')
      0061          WRITE(4) NM, LL, DEBNOL(10), NSLOTS, GAMMA
      0062          CLOSE(UNIT=4)
      0063          755   DO 600 1J=JJ, M1
      C GIVE PROGRESS MESSAGE TO TI:
      WRITE(5, 601) J
      0064          601   FORMAT('1', I3= , I3)
      0065          0065
  
```

POP-11 FORTRAN-77 V4.0-1 08:32:51 16-Jul-86

/F77/TR:BLOCKS/MR
ICAGC.FTN;22

TRASH2=.TRUE.
DEBNJ=STAR1+(13-1)*DBINC

HIGH=DEBNJ GE.15.D0

DEBSR(I,J)=DEBNJ

R=10.00***(DEBNJ/10.00)

RHOTS=GAMMA*R*EBNO/(GAMMA*R+EBNO)

RHOT=R*RHOTS/FLL

C EVALUATE THE PROBABILITY

CALL PSUBELRHON,RHOT,LL,MM,GAMMA,PESYM)

PE=MORBIT*PESYM

WRITE(6,666) DBSR(I,J),PE

FORMAT(1X,F7.3,5X,1PD12.5)

PALOG(I,J)=LOG10(PE)

OPEN(UNIT=4,FILE=FILENAME,STATUS='OLD',ACCESS='APPEND',

FORM='UNFORMATTED')

WRITE(4) DBSR(I,J), PRLG(I,J)

CLOSE(UNIT=4)

OPEN(UNIT=4,FILE=FILENAME,STATUS='NEW',FORM='UNFORMATTED')

WRITE(4) MM,LL,DEBNL(10),MSLOTS,GAMMA,DRSJR,PRLOG

CLOSE(UNIT=4)

CONTINUE

0086 800 CONTINUE

0087 900 CONTINUE

STOP 'PLEASE PURGE DATA FILES'

END

0089

F - 3

POP-11 FORTRAN-77 V4.0-1 08:33:05 16-Jul-86

/F77/TR:BLOCKS/MR
ICAGC.FTN;22

SUBROUTINE GETINJ,START,DBINCR

C INTERACTIVE INPUT OF PARAMETERS FOR RUN

CHARACTER*9 FIELD,BLANK9

COMMON /INPUTS/ DEBNL(NL,IN,

COMMON /SIZE/ NO,NL,NG

DIMENSION D6(31), DSNR(5,4)

DATA DS / .00100, .00200, .00500,

0100, .0200, .0500,

.100, .200, .500, 1.00, 21+0.00/

DATA DSNR /13.3524700, 12.313300, 10.9444300, 0.00, 0.00,

10.60657200, 9.628400, 8.3524800, 0.00, 0.00,

9.0940100, 8.168000, 6.97199500, 0.00, 0.00,

8.0783500, 7.199600, 6.06964600, 0.00, 0.00/

DATA BLANK9,/,

WRITE(5,33)

FORMAT('BITS/SYMBOL (K) [2]: ',\\$)

READ(5,31)K

IF(K.EQ.0)K=2

NN=2**K

0009

CONTINUE

0010 32

FORMAT(' HOW MANY EB/M0? [1]: ',\\$)

0011 33

FORMAT(12)

0012 IF(M0.EQ.0)M0=1

0013 DO 7 IN=1,NN

0014 IF(K.LE.4) THEN

0015 1

WRITE(5,2)

FORMAT(12)

0016 2

FORMAT(' HOW MANY EB/M0? [1]: ',\\$)

0017 0018 3

FORMAT(12)

0019 IF(M0.EQ.0)M0=1

0020 DO 7 IN=1,NN

0021 IF(K.LE.4) THEN

0022 0023

FORMAT(A9)

0024 ELSE

0025 0.00

0026 END IF

0027 4

WRITE(5,5)IN,DO

FORMAT(' EB/M0(',I2,') [',F9.6,']: ',\\$)

0028 READ(5,6)FIELD

0029 6

FORMAT(F9.6)

0030 IF(FIELD.EQ.BLANK9) THEN

0031 DEBNL(IN)=DO

0032 ELSE

0033 DECODE(9,61,FIELD)DEBNL(IN)

0034 61

FORMAT(F9.6)

0035 7

CONTINUE

0036 15

WRITE(5,16)

FORMAT(' HOW MANY L? [4]: ',\\$)

0037 16

FORMAT(12)

0038 17

IF(NL.EQ.0)NL=4

0039 0040 0041

0042 18

WRITE(5,19)IN,IN

FORMAT(' L(',I1,') [',I1,']: ',\\$)

0043 19

READ(5,3)LLIST(IN)

0044

PDP-11 FORTRAN-77 V4.0-1 08:33:05 16-Jul-86
 ICAGC.FTN;22 /F77/TR:BLOCKS/MR

```

0045 IF(LLIST(IN).EQ.0)LLIST(IN)=1
  CONTINUE
0046 21
0047 22 WRITE(5,23)
0048 23 FORMAT(5,23)
  READ(5,24)MSLOTS
0049 24 FORMAT(15)
  IF(MSLOTS.EQ.0)MSLOTS=2400
0050 25 WRITE(5,26)
0051 26 FORMAT(' HOW MANY GAMMA? [10]: ',\$)
  READ(5,3)NG
0052 27 IF(NG.EQ.0)NG=10
0053 28 DO 31 IN=1,NG
  WRITE(5,29)IN,DG(IN)
0054 29 FORMAT(15,10)
  READ(5,30)GAMLST(IN)
0055 30 FORMAT(15,8)
  IF(GAMLST(IN).EQ.0.00)GAMLST(IN)=DG(IN)
  CONTINUE
0056 31 WRITE(5,32)
0057 32 FORMAT(15,12,') [' ,1P08.1,']; ',\$)
  READ(5,33)N
0058 33 FORMAT(15,1)
  IF(N.GT.0)N=N-1
0059 34 FORMAT(I3)
  IF(NJ.EQ.0)NJ=1
0060 35 IF(NJ.LT.0 .OR. NJ.GT.128)GOTO 32
  WRITE(5,41)
0061 36 FORMAT(15,1)
  STARTING VALUE FOR EB/NJ (DB) [0,1]: ',\$)
0062 37 READ(5,42,ERR=40) START
0063 38 FORMAT(F6.3)
0064 39 IF(NJ.EQ.1) RETURN
0065 40 WRITE(5,36)
0066 41 FORMAT(15,1)
  READ(5,37,ERR=35) DBINC
0067 42 FORMAT(F6.3)
  IF(DBINC.EQ.0.) DBINC=5.
  RETURN
0068 43
0069 44
0070 45
0071 46
0072 47
0073 48
0074 49
0075 50
0076 51
0077 52
0078 53
0079 54
0080 55

```

Page 5.

PDP-11 FORTRAN-77 V4.0-1 08:33:15 16-Jul-86
 ICAGC.FTN;22 /F77/TR:BLOCKS/MR

```

0001      C
          C COMPUTE UNCONDITIONAL ERROR PROBABILITY
          C
0002      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          PE=0.00
          DBINC=1.0D-GAMMA
          DO 100 LJ=0,11
0003      ALAM=2.0D*(LL-1)*RNORM*2.0D*1.0D*RNORM
0004      IF((ONG.NE.0.00 .OR. (ONG.EQ.0.00 .AND. LJ.EQ.LL)) THEN
0005          CALL PSI1(ALAM,M,LJ,PECON)
0006          PE=PE+DBINC*(LL,LJ)*DXI(GAMMA,LJ)*DXI(ONG,LJ-L1)*PECON
0007      END IF
0008      CONTINUE
0009      RETURN
0010
0011
0012
0013

```

Page 6.

F-4

PDP-11 FORTRAN-77 V4.0-1 08:33:17 16-Jul-86 Page 7

1CAGC.FTN;22 /F77/TR:BLOCKS/MR

0001 SUBROUTINE PSL1(ALAM,MN,LL,PE)

C COMPUTE CONDITIONAL ERROR PROBABILITY

C

```

0002      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003      DIMENSION C(0:50)
0004      PE=0.0D0
0005      POWERI=-1.0D0
0006      DO 900 K=1,MN-1
0007      FK=K
0008      POWERI=POWERI*(MN-FK)/FK
0009      COEFK=PCWER1/DXI(FK+1.0D0,LL)
0010      IRU=K*(LL-1)
0011      IF (IRU.GT.50) STOP 'C ARRAY TOO SMALL'
0012      FK1=K+1
0013      AL=-FK*ALAM+0.500/FK1
0014      ARG=-AL*AM+0.500/FK1
0015      SUMR=0.0D0
0016      DO 800 IR=0,IRU
0017      R=IR

```

C BUILD UP THE JCP MILLER COEFFICIENT

C

```

0018      IF (IR.EQ.0) THEN
0019          C(0)=1.0D0
0020      ELSE
0021          MUP=MIMO(IR,MAX0(1,LL-1))
0022          SEE=0.0D0
0023          DO 100 N=1,MUP
0024              SEE=SEE+DBINCO(IR,N)*((K+1)**N-IR)*C(IR-N)
0025          CONTINUE
0026          C(IR)=SEE/R
0027      END IF
0028      GL=GENLAG(LL-1,IR,ARG)
0029      Y=DEXP(4X+DLG(GL))
0030      TERM=C(IR)**Y/DXI(FK1,IR)
0031      SUMR=SUMR+TERM
0032      CONTINUE
0033      PE=PE+COEFK*SUMR
0034      CONTINUE
0035      RETURN
0036 800

```

PDP-11 FORTRAN-77 V4.0-1 08:33:22 16-Jul-86 Page 8

1CAGC.FTN;22 /F77/TR:BLOCKS/MR

0001 DOUBLE PRECISION FUNCTION GENLAG(ALFA,N,X)

C GENERALIZED LAGUERRE POLYNOMIALS

C

```

0002      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003      TERM=DBINCO(N+1,ALFA,N)
0004      SUM=TERM
0005      IF (N.EQ.0) GOTO 200
0006      DO 100 M=1,N
0007          TERM=TERM*((X/N)**((N-M+1.0D0)/(N-M+1)))
0008          SUM=SUM+TERM
0009      CONTINUE
0010      100
0011      200
0012      RETURN

```

C

APPENDIX F
COMPUTER PROGRAM FOR
NUMERICAL COMPUTATIONS FOR THE
ANY-CHANNEL-JAMMED ADAPTIVE GAIN CONTROL RECEIVER

The following pages contain the source code listing for the FORTRAN-77 program used for numerical computations of the performance of the any-channel-jammed adaptive gain control receiver for FH/RMFSK.



Page 1

गोपनीय

PDP-11 FORTRAN-77 v4.0-1 16:57:23 15-Jul-86
ACJAGC.FTH;36 /F77/TR:@BLOCKS/MR Page 2

```

0034      WRITE(5,733) FNAME
0035      FORMAT(' WORKING ON FILE ',A13)
0036      OPEN(UNIT=4,FILE=FNAME,STATUS='OLD',FORM='UNFORMATTED',
0037      $           ERR=750)
0038      C HAVE AN EXISTING FILE. READ TO SEE HOW FAR WE GOT BEFORE
0039      C
0040      1300     READ(4,MMIN,LLIN,EBIN,MSIN, GAMIN
0041      $           IF(MMIN.NE.MM .OR. LLIN.NE.LL .OR. EBIN.NE.EBIN)
0042      $           .OR. GAMIN.NE.GAMMA .OR. MSIN.NE.MSLOTS)
0043      $           STOP 'FILE SYNC ERROR OR CORRUPTED FILE'
0044      JJ=0
0045      740       JJ=JJ+1
0046      READ(4,END=742) DBSR(JJ), PRLOG(JJ)
0047      GOTO 740
0048      CLOSE(UNIT=4)
0049      GOTO 755
0050      735       OPEN(UNIT=3,FILE=GNAME,STATUS='OLD',FORM='UNFORMATTED',
0051      $           READONLY,ERR=770)
0052      3939     WRITE(5,3939)
0053      FORMAT(' READING EXISTING EVENT FILE')
0054      READ(31,D1DSUB,MUSED,6000
0055      CLOSE(UNIT=3)
0056      GOTO 777
0057      770       CONTINUE
0058      3938     WRITE(5,3938)
0059      FORMAT(' CREATING EVENT FILE')
0060      CALL GEMPIE(LL,MM,NQ,MSLOTS,GOOD,MATRIX,MLOW,MINC,MUP,PIN,
0061      $           D1DSUB,MUSED)
0062      777       OPEN(UNIT=3,FILE=GNAME,STATUS='NEW',FORM='UNFORMATTED')
0063      WRITE(3,D1DSUB,MUSED,GOOD
0064      CLOSE(UNIT=3)
0065      IF( NOT GOOD) GOTO 700
0066      DO 600 I=1,J,J,M
0067      C GIVE PROGRESS MESSAGE TO TI:
0068      WRITE(5,601) I,J
0069      FORMAT(' I,J=',I3,' ,',J3)

```

```

POP-11 FORTRAN-77 V4.0-1   16:57:23  15-Jul-86
ACJAGC.FTN;36   /F77/TR:BLOCKS/MR

0068      TRASH2=.TRUE.
0069      DEBNJ=START+(1/J-1)*DB1MC
0070      DB$JN(1,J)=DEBNJ
0071      R=10.0**((DEBNJ/10.00))
0072      RMOTS=GMAMMA*R*(EBNO)/(GMAMMA+R*EBNO)
0073      RMOTS=GMAMMA*R*(EBNO)/(GMAMMA+R*EBNO)
0074      C EVALUATE THE PROBABILITY
0075      CALL PSUBE(RMAM, RMOT, LL, MM, BITS, PESTM,
0076           IPSUB, DEBNJ)
0077      PE=MORBIT(PESTM
0078      WRITE(6,666) DSJR(1,J), PE
0079      FORMAT(IX,F7.3,5X,IP012.5)
0080      PRLOG(1,J)=DL0610(PE)
0081      OPEN(UNIT=4,FILE=FNAME,STATUS='OLD',AC-
0082      FORM='UNFORMATTED')
0083      WRITE(4,444,LL,DSJR(1,J)).PRLOG(1,J)
0084      CLOSE(UNIT=4)
0085      CONTINUE
0086      CONTINUE
0087      CONTINUE
0088      STOP 'PLEASE PURGE DATA FILES'
0089

```

POP-11 FORTRAN-77 V4.0-1 16:57:23 15-Jul-86
ACJAGC.FTN:36 /F77/TR:BLOCKS/MR Page 3

```

Page 4
PDF-11 FORTRAN-77 V4.0-1 16:57:36 15-Jul-86
ACJAGC.FTM;36 /F77/TR:BLOCKS/MR

0001      C          SUBROUTINE GET(IN, START, DBINIC)

          C          C          INTERACTIVE INPUT OF PARAMETERS FOR RUN
          C          C          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          C          CHARACTER*9 FIELD,BLANK9
          COMMON /INPUTS/ DEBNOL(5),LLIST(4),NSLOTS,GMLST(31),K,I
          COMMON /SIZE/ NO_ML,MS
          C          DEFAULT LISTS TEMPORARILY NEEDED ARE IN SHARED STORAGE WITH
          C          THE LARGE CONVOLUTION WORKING ARRAYS
          COMMON /SHARE/ DS(31),DSR(5,4)
          DATA BLANK9/,/
          WRITE(5,33)
          FORMAT(1X,BITS/SYMBOL, (K) [2]: ',$,)
          READ(5,3)K
          IF(K.EQ.0)K=2
          MS=2**K
          WRITE(5,32)
          FORMAT(1X, HOW MANY EB/NO? [1]: ',$,)
          READ(5,3)NO
          FORMAT(1X,2)
          IF(NO.EQ.0)NO=1
          DO 7 IN=1,NO
          IF(K.LE.4) THEN
          DO=DSNR(1N,K)
          ELSE
          DO=0.00
          END IF
          WRITE(5,5)IN,DO
          FORMAT(1X,EB/NO(' ,I2,') ',F9.6,'): ',$,)
          READ(5,6)FIELD
          FORMAT(1A9)
          IF(FIELD.EQ.BLANK9) THEN
          DEBNOL(1N)=DO
          ELSE
          DEBML(1N)DEBML(1N)
          FORMAT(9,6)
          END 1F
          CONTINUE
          WRITE(5,16)
          FORMAT(1X, HOW MANY L? [1]: ',$,)
          READ(5,3)NL
          IF(NL.EQ.0)NL=1
          DO 21 IN=1,NL
          WRITE(5,19)IN
          FORMAT(1X, 'L' ,I1, ') ' ,I4): ',$,)
          READ(5,3)LLIST(IN)
          IF(LLIST(IN).EQ.0)LLIST(IN)=4
          CONTINUE
          WRITE(5,23)
          FORMAT(1X, HOPPING SLOTS? [2400]: ',$,)
          READ(5,24)NSLOTS
          33
          32
          31
          30
          29
          28
          27
          26
          25
          24
          23
          22
          21
          20
          19
          18
          17
          16
          15
          14
          13
          12
          11
          10
          9
          8
          7
          6
          5
          4
          3
          2
          1

```

PDP-11 FORTRAN-77 V4.0-1 16:57:36 15-Jul-86
 ACJAGC.FTN;36 /F77/TR:BLOCKS/NR

PDP-11 FORTRAN-77 V4.0-1 16:57:45 15-Jul-86
 ACJAGC.FTN;36 /F77/TR:BLOCKS/NR

```

0048   24      FORMAT(15)
0049     IF(NSLOTS.EQ.0)NSLOTS=2400
0050   25      WRITE(15,26)
0051   26      FORMAT(' HOW MANY GAMMA? ',I3): ',$'
0052   27      READ(15,3)NG
0053     IF(NG.EQ.0)NG=3
0054   28      DO 31 1M=1,NG
0055   29      WRITE(15,29)IN,DE(IN)
0056   30      FORMAT('  GAMMA( ',I2,' ) [',1PD8.1,',]: ',$')
0057   31      READ(15,30,ERR=28)GAMLIST(IN)
0058   32      FORMAT(15,8)
0059   33      IF(GAMLIST(IN).EQ.0.0)GAMLIST(IN)=DG(IN)
0060   34      CONTINUE
0061   35      WRITE(15,39)
0062   36      FORMAT(' HOW MANY EB/NJ? ',I26]: ',$')
0063   37      READ(15,34,ERR=38)NJ
0064   38      FORMAT(13)
0065   39      IF(NJ.EQ.0) NJ=126
0066   40      IF(NJ.LT.0 .OR. NJ.GT.126) 6070 32
0067   41      WRITE(15,41)
0068   42      FORMAT(' STARTING VALUE FOR EB/NJ (DB) [0.]: ',$)
0069   43      READ(15,42,ERR=40) START
0070   44      FORMAT(F6.3)
0071   45      IF(NJ.EQ.1) RETURN
0072   46      WRITE(15,36)
0073   47      FORMAT(' DB INCREMENT FOR EB/NJ [.4]: ',$)
0074   48      READ(15,37,ERR=35) DBINC
0075   49      FORMAT(F6.3)
0076   50      IF(DBINC.EQ.0.) DBINC=0.400
0077   51      RETURN
0078   52      END
  T-4

```

```

0001      5      SUBROUTINE PSUBE(RHOT,LL,M,BITS,
0002          PE,O,DSUB,MUSED,PRERR,IPSUB,EBNJI)
0003      C      COMPUTE UNCONDITIONAL ERROR PROBABILITY
0004      C
0005      C      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0006      C      INTEGER JAML(0:8)
0007      C      LOGICAL TRASH1,TRASH2
0008      C      VIRTUAL PRERR(625),IPSUB(625)
0009      C      VIRTUAL D(625),IDSUB(625)
0010      C      COMMON /RESET/ TRASH1,TRASH2
0011      C      PE=0.00
0012      C      DO 199 LTHJ=0,LL
0013      C      JAML(0)*LTHJ
0014      C      FLAG TO MAKE SURE TRASH1 IS TRUE FOR FIRST TIME AROUND
0015      C      JAML(1)==1
0016      C      IF(L.GT.2) THEN
0017      C      JAML(1)=JAML(1-1)
0018      C      ELSE
0019      C      JAML(1)=0
0020      C      END IF
0021      C      CONTINUE
0022      C      CALL EVENT(LL,M,JAML,PIE,D,DSUB,MUSED)
0023      C      IF(PIE.EQ.0.00) GOTO 190
0024      C      TRANSFORM TO EQUIVALENT LINEAR RECEIVER CONDITIONAL PROBABILITY
0025      C      RHOT0=RHOT1
0026      C      IF(JAML(1)==JAML(1-1)) THEN
0027      C      ELSE
0028      C      RHOT1=RHOT
0029      C      END IF
0030      C      IF(RHOT0.NE.RHOT1) TRASH2=.TRUE.
0031      C      JAM1=JAM1
0032      C      DO 105 1=1,M
0033      C      JAM1=LL-JAM1(0)+JAM1(1)
0034      C      CONTINUE
0035      C      IF(JAM1.NE.JAM(1)) TRASH1=.TRUE.
0036      C      UCX=RHON/RHOT
0037      C      IF(RHOT1.NE.RHON) THEN
0038      C      EBNJ1=LL*RHON+RHOT1/(BITS*(RHOT1-RHOM))
0039      C      ELSE
0040      C      EBNJ1=LL*RHON/BITS
0041      C      END IF

```

PDP-11 FORTRAN-77 V4.0-1 16:57:45 15-Jul-86 Page 7
 ACJAGC.FTN;36 /F77/TR:BLOCKS/MR

```

C EMPIRICAL SWITCH-OVER POINT FOR COMPUTATIONAL METHOD
C TO AVOID ROUND-OFF AND TRUNCATION ERRORS
  IF(EBN1.GT.31.6 .OR. EBN1.LE.-40.00 .OR.
     (RHON.GE.30.00 .AND. EBN1.LE.25.00)
     .OR. UCK.LT.-1.0100) THEN
    CALL PSEL1(JAM,L1,M,RHOM,RMOT1,UCK,PROB)
  ELSE
    CALL PSEL2(JAM,L1,M,RHOM,RMOT1,UCK,PROB)
    TRASH1=.TRUE.
    TRASH2=.TRUE.
  END IF
  PE=PE+PROB*PIE
  DO 195 I=2,M
  ITER=M+2-I
  JAML(ITER)=JAML(ITER)+1
  IF(JAML(ITER).LE.LTH) GOTO 100
  CONTINUE
  195 CONTINUE
  199 RETURN
  END

```

PDP-11 FORTRAN-77 V4.0-1 16:57:51 15-Jul-86 Page 8
 ACJAGC.FTN;36 /F77/TR:BLOCKS/MR

```

SUBROUTINE EVENT(LL,M,JAML,PIE,O,ISUB,MUSED)
0001  C SUBROUTINE TO LOOK UP EVENT PROBABILITY FROM STORED ARRAY
C
C IMPLICIT DOUBLE PRECISION(A-M,0-Z)
LOGICAL*1 STORE,MUNE
DIMENSION JAML(9),LUP(9)
VIRTUAL O1(625),ISUB(625)
DIMENSION LOW(9)
DATA STORE/.FALSE./,LOW/9*0/
C SET UP ARRAY DESCRIPTION D(0:LL,...,0:LL) WITH M+1 DIMENSIONS
0002 0003
0004 0005
0005 0006
0006 0007
0007 0008
0008 0009
0009 0010
0010 1
1 1 CONTINUE
C COMPUTE LINEAR EQUIVALENT SUBSCRIPT FOR JAMMING EVENT
CALL LOCN(M+1,LOW,LUP,JAML,ISUB)
0011 0012
0012 0013
0013 0014
0014

```

```

0001      SUBROUTINE GEMPE(LL,MM,MQ,MSLOTS,GOOD,MATRIX,MLOW,MINC,
0002          MUP,PIE,D,IDSUB,MUSED)
0003      C SUBROUTINE TO GENERATE EVENT PROBABILITIES
0004      C
0005      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0006      LOGICAL1 60,STORE,NONE,GOOD
0007      DIMENSION MATRIX(LL,MM),MLOW(LL,MM),MINC(LL,MM),MUP(LL,MM),
0008          PIE(0:MM),IWORK(0:8),DWORK(0:8),LUPWORK(0:8)
0009      VIRTUAL D(625),IDSUB(625)
0010      C STORE=.FALSE. -> DON'T STORE ZERO ELEMENTS OF SPARSE ARRAY.
0011      GOOD=.TRUE.
0012      IF(MQ.LE.0) THEN
0013          GOOD=.FALSE.
0014      END IF
0015      DO 90 I=0,MM
0016          CALL PRIMOP(I,MM,MQ,MSLOTS,A)
0017          PIE(I)=A
0018          MINC(I,J)=1
0019          MUP(I,J)=1
0020          LLOWR(I,J)=1
0021          CONTINUE
0022          MUSED=0
0023          DO 98 J=1,MM
0024              LOMARR(J)=0
0025              LUPARR(J)=LL
0026          CONTINUE
0027          CALL MINIT(MATRIX,MLOW,LL,MM)
0028          CONTINUE
0029          C FORM COLUMN SUMS AND COUNT JUMPED HOPS AND COMPUTE P(EVENT)
0030          P=1.00
0031          DO 101 I=1,LL
0032              K=0
0033              DO 100 J=1,MM
0034                  K=K+MATRIX(I,J)
0035              CONTINUE
0036              IF(K.NE.0) LTHJ=LTHJ+1
0037              P=P*PIE(K)
0038              CONTINUE
0039              C FORM JUMPING EVENT VECTOR
0040              IWORK(1)=LTHJ
0041              DO 102 J=1,MM
0042                  IWORK(J)=IWORK(J)+MATRIX(I,J)
0043              CONTINUE
0044          102 CONTINUE

```

```

0045      C SORT NON SIGNAL CHANNELS
0046          DO 103 I=2,MM-1
0047              DO 103 J=I+1,MM
0048                  IF(IWORK(J).LT.IWORK(I)) THEN
0049                      TTEMP=IWORK(I)
0050                      IWORK(I)=IWORK(J)
0051                      IWORK(J)=TTEMP
0052                  CONTINUE
0053                  CALL LOCN(MP+1,LDMARK,LUPMARK,IWORK,1SUB)
0054                  CALL LOOKUP(DOUT,D,1DSUB,MUSED,625,1SUB,STORE,MONE)
0055                  DOIT=DOUT+p
0056                  CALL PUTIN(DOUT,D,1DSUB,MUSED,625,1SUB,IERR,STORE)
0057                  IF(IERR.NE.0) STOP 'TOO MANY EVENTS'
0058                  CALL MINIT(MATRIX,MLOW,MUP,MINC,LL,MM,B0)
0059                  IF(B0) GOTO 999
0060                  RETURN
0061
0062      C ITERATE MATRIX-INDEX LOOP
0063
0064      103
0065
0066      9999
0067
0068      9999
0069
0070      9999
0071
0072      9999
0073
0074      9999
0075
0076      9999
0077
0078      9999
0079
0080      9999
0081
0082      9999
0083
0084      9999
0085
0086      9999
0087
0088      9999
0089
0090      9999
0091
0092      9999
0093
0094      9999
0095
0096      9999
0097
0098      9999
0099
0100      9999
0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111
0112
0113
0114
0115
0116
0117
0118
0119
0120
0121
0122
0123
0124
0125
0126
0127
0128
0129
0130
0131
0132
0133
0134
0135
0136
0137
0138
0139
0140
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154
0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168
0169
0170
0171
0172
0173
0174
0175
0176
0177
0178
0179
0180
0181
0182
0183
0184
0185
0186
0187
0188
0189
0190
0191
0192
0193
0194
0195
0196
0197
0198
0199
0200
0201
0202
0203
0204
0205
0206
0207
0208
0209
0210
0211
0212
0213
0214
0215
0216
0217
0218
0219
0220
0221
0222
0223
0224
0225
0226
0227
0228
0229
0230
0231
0232
0233
0234
0235
0236
0237
0238
0239
0240
0241
0242
0243
0244
0245
0246
0247
0248
0249
0250
0251
0252
0253
0254
0255
0256
0257
0258
0259
0260
0261
0262
0263
0264
0265
0266
0267
0268
0269
0270
0271
0272
0273
0274
0275
0276
0277
0278
0279
0280
0281
0282
0283
0284
0285
0286
0287
0288
0289
0290
0291
0292
0293
0294
0295
0296
0297
0298
0299
0300
0301
0302
0303
0304
0305
0306
0307
0308
0309
0310
0311
0312
0313
0314
0315
0316
0317
0318
0319
0320
0321
0322
0323
0324
0325
0326
0327
0328
0329
0330
0331
0332
0333
0334
0335
0336
0337
0338
0339
0340
0341
0342
0343
0344
0345
0346
0347
0348
0349
0350
0351
0352
0353
0354
0355
0356
0357
0358
0359
0360
0361
0362
0363
0364
0365
0366
0367
0368
0369
0370
0371
0372
0373
0374
0375
0376
0377
0378
0379
0380
0381
0382
0383
0384
0385
0386
0387
0388
0389
0390
0391
0392
0393
0394
0395
0396
0397
0398
0399
0400
0401
0402
0403
0404
0405
0406
0407
0408
0409
0410
0411
0412
0413
0414
0415
0416
0417
0418
0419
0420
0421
0422
0423
0424
0425
0426
0427
0428
0429
0430
0431
0432
0433
0434
0435
0436
0437
0438
0439
0440
0441
0442
0443
0444
0445
0446
0447
0448
0449
0450
0451
0452
0453
0454
0455
0456
0457
0458
0459
0460
0461
0462
0463
0464
0465
0466
0467
0468
0469
0470
0471
0472
0473
0474
0475
0476
0477
0478
0479
0480
0481
0482
0483
0484
0485
0486
0487
0488
0489
0490
0491
0492
0493
0494
0495
0496
0497
0498
0499
0500
0501
0502
0503
0504
0505
0506
0507
0508
0509
0510
0511
0512
0513
0514
0515
0516
0517
0518
0519
0520
0521
0522
0523
0524
0525
0526
0527
0528
0529
0530
0531
0532
0533
0534
0535
0536
0537
0538
0539
0540
0541
0542
0543
0544
0545
0546
0547
0548
0549
0550
0551
0552
0553
0554
0555
0556
0557
0558
0559
0550
0551
0552
0553
0554
0555
0556
0557
0558
0559
0560
0561
0562
0563
0564
0565
0566
0567
0568
0569
0560
0561
0562
0563
0564
0565
0566
0567
0568
0569
0570
0571
0572
0573
0574
0575
0576
0577
0578
0579
0580
0581
0582
0583
0584
0585
0586
0587
0588
0589
0580
0581
0582
0583
0584
0585
0586
0587
0588
0589
0590
0591
0592
0593
0594
0595
0596
0597
0598
0599
0590
0591
0592
0593
0594
0595
0596
0597
0598
0599
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0610
0611
0612
0613
0614
0615
0616
0617
0618
0619
0620
0621
0622
0623
0624
0625
0626
0627
0628
0629
0630
0631
0632
0633
0634
0635
0636
0637
0638
0639
0640
0641
0642
0643
0644
0645
0646
0647
0648
0649
0640
0641
0642
0643
0644
0645
0646
0647
0648
0649
0650
0651
0652
0653
0654
0655
0656
0657
0658
0659
0650
0651
0652
0653
0654
0655
0656
0657
0658
0659
0660
0661
0662
0663
0664
0665
0666
0667
0668
0669
0660
0661
0662
0663
0664
0665
0666
0667
0668
0669
0670
0671
0672
0673
0674
0675
0676
0677
0678
0679
0680
0681
0682
0683
0684
0685
0686
0687
0688
0689
0680
0681
0682
0683
0684
0685
0686
0687
0688
0689
0690
0691
0692
0693
0694
0695
0696
0697
0698
0699
0690
0691
0692
0693
0694
0695
0696
0697
0698
0699
0700
0701
0702
0703
0704
0705
0706
0707
0708
0709
0700
0701
0702
0703
0704
0705
0706
0707
0708
0709
0710
0711
0712
0713
0714
0715
0716
0717
0718
0719
0710
0711
0712
0713
0714
0715
0716
0717
0718
0719
0720
0721
0722
0723
0724
0725
0726
0727
0728
0729
0720
0721
0722
0723
0724
0725
0726
0727
0728
0729
0730
0731
0732
0733
0734
0735
0736
0737
0738
0739
0740
0741
0742
0743
0744
0745
0746
0747
0748
0749
0740
0741
0742
0743
0744
0745
0746
0747
0748
0749
0750
0751
0752
0753
0754
0755
0756
0757
0758
0759
0750
0751
0752
0753
0754
0755
0756
0757
0758
0759
0760
0761
0762
0763
0764
0765
0766
0767
0768
0769
0760
0761
0762
0763
0764
0765
0766
0767
0768
0769
0770
0771
0772
0773
0774
0775
0776
0777
0778
0779
0780
0781
0782
0783
0784
0785
0786
0787
0788
0789
0780
0781
0782
0783
0784
0785
0786
0787
0788
0789
0790
0791
0792
0793
0794
0795
0796
0797
0798
0799
0790
0791
0792
0793
0794
0795
0796
0797
0798
0799
0800
0801
0802
0803
0804
0805
0806
0807
0808
0809
0800
0801
0802
0803
0804
0805
0806
0807
0808
0809
0810
0811
0812
0813
0814
0815
0816
0817
0818
0819
0810
0811
0812
0813
0814
0815
0816
0817
0818
0819
0820
0821
0822
0823
0824
0825
0826
0827
0828
0829
0820
0821
0822
0823
0824
0825
0826
0827
0828
0829
0830
0831
0832
0833
0834
0835
0836
0837
0838
0839
0840
0841
0842
0843
0844
0845
0846
0847
0848
0849
0840
0841
0842
0843
0844
0845
0846
0847
0848
0849
0850
0851
0852
0853
0854
0855
0856
0857
0858
0859
0850
0851
0852
0853
0854
0855
0856
0857
0858
0859
0860
0861
0862
0863
0864
0865
0866
0867
0868
0869
0860
0861
0862
0863
0864
0865
0866
0867
0868
0869
0870
0871
0872
0873
0874
0875
0876
0877
0878
0879
0880
0881
0882
0883
0884
0885
0886
0887
0888
0889
0880
0881
0882
0883
0884
0885
0886
0887
0888
0889
0890
0891
0892
0893
0894
0895
0896
0897
0898
0899
0890
0891
0892
0893
0894
0895
0896
0897
0898
0899
0900
0901
0902
0903
0904
0905
0906
0907
0908
0909
0900
0901
0902
0903
0904
0905
0906
0907
0908
0909
0910
0911
0912
0913
0914
0915
0916
0917
0918
0919
0910
0911
0912
0913
0914
0915
0916
0917
0918
0919
0920
0921
0922
0923
0924
0925
0926
0927
0928
0929
0920
0921
0922
0923
0924
0925
0926
0927
0928
0929
0930
0931
0932
0933
0934
0935
0936
0937
0938
0939
0940
0941
0942
0943
0944
0945
0946
0947
0948
0949
0940
0941
0942
0943
0944
0945
0946
0947
0948
0949
0950
0951
0952
0953
0954
0955
0956
0957
0958
0959
0950
0951
0952
0953
0954
0955
0956
0957
0958
0959
0960
0961
0962
0963
0964
0965
0966
0967
0968
0969
0960
0961
0962
0963
0964
0965
0966
0967
0968
0969
0970
0971
0972
0973
0974
0975
0976
0977
0978
0979
0980
0981
0982
0983
0984
0985
0986
0987
0988
0989
0980
0981
0982
0983
0984
0985
0986
0987
0988
0989
0990
0991
0992
0993
0994
0995
0996
0997
0998
0999
0990
0991
0992
0993
0994
0995
0996
0997
0998
0999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1380
1381
1
```

PDP-11 FORTRAN-77 V4.0-1 16:58:07
ACJAGC.FTN;36 /F77/TR:BLOCKS;WR

Page 11

0001 SUBROUTINE PUTIN(CIN,C,ICSUB,MUSE,MAX,K,IERR,STORE)

C THIS SUBROUTINE INSERTS AN ELEMENT INTO A SPARSE ARRAY FOR
C WHICH ONLY THE NON-ZERO ELEMENTS ARE KEPT IN STORAGE IF
C THE SWITCH STORE IS .TRUE.

C THE DOUBLE PRECISION VALUE CIN IS STORED AS C(K) WHERE
C THE AUXILIARY ARRAY ICSUB IS USED TO KEEP TRACK OF THE
C SUBSCRIPTS OF THE CORRESPONDING ENTRIES IN C.
C LONG (4-BYTE) INTEGERS ARE USED TO ACCOMMODATE LARGE
C SUBSCRIPT VALUES FOR THE SPARSE ARRAY C.

C USAGE:

C LOGICAL*1 STORE
C DOUBLE PRECISION C,CIN
C VIRTUAL ICSUB(MAX),C(MAX)
C CALL PUTIN(CIN,C,ICSUB,MUSE,MAX,K,IERR,STORE)

C WHERE

C CIN = VALUE OF ELEMENT TO STORE
C C = ARRAY IN WHICH NON-ZERO VALUES ARE ACTUALLY STORED

C ICSUB = AUXILIARY ARRAY FOR ACTUAL SUBSCRIPT VALUES
C MUSE = NUMBER OF ELEMENTS OF C CURRENTLY OCCUPIED

C MAX = SIZE OF ARRAY C
C I = ERROR RETURN CODE, 0 IF NO ERROR OR 1 IF THERE IS
C J ROOM AVAILABLE IN C

C S ..E = .TRUE. TO STORE ZEROES EXPLICITLY, ELSE .FALSE.

C NOTE: IF CIN=0 AND THE SUBSCRIPT K IS FOUND IN ICSUB, THEN
C THE ELEMENT IS DELETED BY SHIFTING DOWNWARD ALL
C FOLLOWING ELEMENTS OF THE ARRAY

C PROGRAMMER: ROBERT H. FRENCH DATE: 11 JANUARY 1,34

IMPLICIT DOUBLE PRECISION(A-H,O-Z)

VIRTUAL ICSUB(MAX),C(MAX)

LOGICAL*1 STORE

IERR=0

IF(STORE) GOTO 5

IF(CIN EQ.0 GO TO 50

IF(MUSE EQ.0) GOTO 20

DO 10 I=1,MUSE

IF(ICSUB(I).NE.K) GOTO 10

C(I)=CIN

RETURN

CONTINUE

IF(MUSE LT. MAX) GOTO 20

IERR=1

RETURN

MUSE=MUSE+1

ICSUB(MUSE)=K

C(MUSE)=CIN

RETURN

PDP-11 FORTRAN-77 V4.0-1 16:58:00
ACJAGC.FTN;36 /F77/TR:BLOCKS;WR

Page 12

0002 DO 40 I=1,MUSE

J=1

IF(ICSUB(I).EQ.K) GOTO 50

CONTINUE

RETURN

C REMOVE THE ZEROED ELEMENT AND BUMP COUNT OF ENTRIES USED

C

DO 60 I=J,MUSE-1

ICSUB(I)=ICSUB(I+1)

C(I)=C(I+1)

CONTINUE

MUSE=MUSE-1

RETURN

END

0026 DO 60 I=J,MUSE-1

ICSUB(I)=ICSUB(I+1)

C(I)=C(I+1)

CONTINUE

MUSE=MUSE-1

RETURN

0031 END

0032

```

0001      SUBROUTINE LOOKUP(COUT,C,ICSUB,N,MMAX,K,STORE,MONE)
C
C THIS SUBROUTINE RETRIEVS AN ELEMENT OF A SPARSE ARRAY WHICH
C HAS BEEN STORED COMPACTLY BY STORING ONLY NON-ZERO ELEMENTS.
C
C THE ARRAY IS DOUBLE PRECISION.
C
C USAGE:   VIRTUAL ICSub(1MMAX), C(1MMAX)
C          LOGICAL*1 STORE, MONE
C          DOUBLE PRECISION COUT
C          CALL LOOKUP(COUT,C,ICSub,N,MMAX,K,STORE,MONE)
C
C WHERE
C       COUT = VALUE OF C(K) (OUTPUT FROM SUBROUTINE)
C       C = ARRAY USED TO STORE NON-ZERO ELEMENTS
C       ICSub = AUXILIARY ARRAY TO STORE ACTUAL SUBSCRIPTS
C       N = NUMBER OF ELEMENTS OF C CURRENTLY IN USE
C       MMAX = SIZE OF C
C       K = SUBSCRIPT OF SPARSE ARRAY TO LOOK UP
C
C       STORE = .TRUE. IF ZEROES STORED EXPLICITLY, ELSE FALSE.
C       MONE = .FALSE. IF ZEROES NOT STORED OR ZEROES STORED AND
C                   ELEMENT IS FOUND IN THE STORED ARRAY
C
C       TRUE. IF ZEROES ARE STORED AND THE ELEMENT IS
C       NOT FOUND (OUTPUT QUANTITY)
C
C PROGRAMMER: ROBERT H. FRENCH
C DATE: 11 JANUARY 1984

```

```

0002      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003      VIRTUAL ICSub(1MMAX), C(1MMAX)
0004      LOGICAL*1 STORE, MONE
0005      MONE=.FALSE.
0006      DO 10 I=1,N
0007      IF(ICSub(I).NE.K)GOTO 10
0008      COUT=C(I)
0009      RETURN
0010      10      CONTINUE
0011      IF(STORE) THEN
0012      MONE=.TRUE.
0013      ELSE
0014      COUT=0.
0015      END IF
0016      RETURN
0017

```

```

PDP-11 FORTRAN-77 V4.0-1      16:58:07    15-Jul-86      Page 15
ACJAGC.FTN;36      /F77/TR:BLOCKS/MR

0001   SUBROUTINE PR1HOP(K,JAM,KM,KQ,KN,AIN)
0002   C THIS SUBROUTINE COMPUTES THE EVENT PROBABILITY FOR ALL
0003   C POSSIBLE JUMPING PATTERNS WITH NON-ZERO PROBABILITY FOR
0004   C L=1 HOP SYMBOL FOR RMFSK/FH IN PBnj
0005   C
0006   C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0007   AIN=0.00
0008   C IF THIS IS AN IMPOSSIBLE CASE, RETURN WITH RESULT T = 0.0
0009   IF(KJAM.GT.MINGO(KM)) RETURN
0010   KMAX=K,JAM-1
0011   LPMAX=IN-K,JAM-1
0012   JPMAX=KM-1
0013   JMAX=MAX0(KMAX,LPMAX,JPMAX)
0014   PROD=1.D0
0015   Q=KQ
0016   DIFFNQ=KN-KQ
0017   EN=KN
0018   DO 100 LOOP=0,IMAX
0019   F=LOOP
0020   IF(LOOP.LE.KMAX) PROD=PROD*(Q-F)
0021   IF(LOOP.LE.JPMAX) PROD=PROD/(EN-F)
0022   IF(LOOP.LE.LPMax) PROD=PROD*(DIFFNQ-F)
0023   CONTINUE
0024   AIN=PROD
0025   RETURN
0026 END

```

```

PDP-11 FORTRAN-77 V4.0-1      16:58:09    15-Jul-86      Page 16
ACJAGC.FTN;36      /F77/TR:BLOCKS/MR

```

```

0001   BLOCK DATA
0002   C INITIALIZE SHARED CONSTANTS
0003   C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0004   C COMMON /SHARE/ D6(31),DSMR(5,4)
0005   C COMMON /SHAREZ/ LM(4),LINC(4)
0006   C DEFAULT LISTS FOR INTERACTIVE PARAMETER INPUTS
0007   C ARE SHARED WITH LARGE WORKING STORAGE ARRAYS SINCE THEY
0008   C MAY BE DESTROYED ONCE THE INPUT PARAMETERS ARE SET UP
0009   DATA D6 / .00100, .00200, .00500,
0010   C
0011   DATA DSMR /13.35247D0, 12.31333D0, 10.94443D0, 0.00, 0.00,
0012   C
0013   DATA DSZR /10.606572D0, 9.62848D0, 8.35248D0, 0.00, 0.00,
0014   C
0015   DATA D9 /9.09401D0, 8.16900D0, 6.971995D0, 0.00, 0.00,
0016   C
0017   DATA D10 /8.07835D0, 7.1996D0, 6.06946D0, 0.00, 0.00/
0018   C FREQUENTLY NEEDED CONSTANT ARRAYS AND SCALARS
0019   DATA LM/4*D0/,LINC/4*D0/
0020   END

```

```

PDP-11 FORTRAN-77 V4.0-1   16:58:10  15-Jul-86
ACMGC.FTN;36   /F77/TR:BLOCKS/ER
0001      SUBROUTINE PSEL1((JSUB,L1,L2,M1,M2,RHOM,RHOT,UCK,PROB)
0002
0003      C RANDOM NF SK/FH IN PARTIAL BAND TONE JAMMING,
0004      C GIVEN A JAMMING EVENT
0005
0006      C JSUB - JAMMING EVENT VECTOR
0007      C L1 - NUMBER OF HOPS/SYMBOL
0008      C MN - ALPHABET SIZE
0009      C PROB - RESULTING CONDITIONAL ERROR PROBABILITY
0010
0011      C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0012      C EXTERNAL PGRND, DSU20
0013      C DIMENSION WORK(75). STACK(75). SAVE(75)
0014      C INTEGER JSUB(4)
0015      C INTEGER NCHAN(0:6)
0016      C COMMON /PARMS/ BIGK, LL
0017      C AJ = NONCENTRAL PARAMETER FOR JAMMED HOPS
0018      C AJJ = NONCENTRAL PARAMETER FOR NON-JAMMED HOPS
0019      C NJ = NUMBER OF JAMMED HOPS
0020      C NMJ = NUMBER OF NON-JAMMED HOPS
0021      C COMMON /PUIPAR/ AJ, AJJ, NJ, NMJ
0022      C COMMON /JANCH1/ NCHAN
0023      C LL=LLL
0024      C BIGK=UCK
0025      C NJ=JSUB(1)
0026      C NMJ=LL-NJ
0027      C AJ=Z*DO+AJJ*RHOT
0028      C AJJ=Z*DO+NMJ*RHOT
0029      C ASHIFT=DMAX1(AJ,AJJ)
0030      C SUBTRACT 1 FOR BENEFIT OF RAPID CHI-SQUARE DENSITY CALCULATION
0031      C NJ=NJ-1
0032      C NMJ=NMJ-1
0033
0034      C COUNT NUMBER OF NON-SIGNAL CHANNELS WITH L HOPS JAMMED
0035
0036      DO 1 I=0,LL
0037      NCHAN(I)=0
0038      1 CONTINUE
0039      DO 2 I=2,NM
0040      JSUB=JSUB(1)
0041      NCHAN(JSUB)=NCHAN(JSUB)+1
0042
0043      2 CONTINUE
0044      CALL DLGS15(PGRND,ASHIFT,TAIL)
0045      CALL ARQUAD(0,ASHIFT,BODY,DSU20,PGRND,1,D-10,
0046                  WORK,STACK,75,KODE)
0047
0048      IF(KODE.NE.0) THEN
0049          WRITE(5,3) KODE
0050          FORMAT(' ARQUAD ERROR CODE = ',I1)
0051          STOP 'FATAL ERROR'
0052
0053      END IF
0054      PROB=TAIL+BODY
0055
0056      RETURN
0057      END

```

```

PPF-11 FORTRAN-77 V4.0-1      16:52:13   15-Jul-86
ACJAGC.FTN.36      /F77/TR:BLOCKS/MR

0001      DOUBLE PRECISION FUNCTION PGAND(BETA)
0002      IMPLICIT DOUBLE PRECISION(A,H,O,Z)
0003      LOGICAL TRASH1, TRASH2
0004      VIRTUAL PU1V(2053), PU1T(2053), IPU1T(2053),
0005          FLV(8191),   FLT(8191),   IFLT(8191)
0006      INTEGER ICHAN(0:6)
0007      COMMON /JAMCUT/ ICHAN
0008      COMMON /RESET/ TRASH1, TRASH2
0009      COMMON /PARMS/ BICK, LL
0010      C      AJ = NONCENTRAL PARAMETER FOR JAMMED HOPS
0011      C      ANJ = NUMBER OF JAMMED HOPS
0012      C      NMJ = NUMBER OF NON-JAMMED HOPS
0013      C      COMMON /PU1PAR/ AJ, ANJ, NMJ, NMJ
0014      C      IF (TRASH1) THEN
0015          C      TRASH THE SIGNAL CHANNEL DENSITY TABLES
0016          DO 1 1=1,2053
0017          1      PU1Y(1)=0.00
0018          DO 2 1=1,2053
0019          2      PU1T(1)=0.00
0020          DO 3 1=1,2053
0021          3      IPU1T(1)=0
0022          IFLT(1)=0
0023          TRASH1=.FALSE.
0024          END IF
0025          IF (TRASH2) THEN
0026              C      TRASH THE NON-SIGNAL CHANNEL DENSITY TABLES
0027          PROB=1.00
0028          DO 10 I=0,LL
0029          10     IF (ICHAN(I).NE.0) THEN
0030              C      IF (ICHAN(I).NE.0) THEN
0031                  ISUB=IHASH(BETA,8191)
0032                  JSUB=ISUB
0033                  T=FLT(156B)
0034                  IT=IFT(1SUB)
0035                  IF (T.EQ.0 .AND. IT.EQ.0) THEN
0036                      C      NOT FOUND, COMPUTE IT AND ENTER INTO TABLE
0037                      X=F1(BETA,1)
0038                      FLT(1SUB)=BETA
0039                      IF (1SUB)=IT+1
0040                      ELSE IF (T.NE.BETA .OR. IT.NE.1+1) THEN
0041                          C      NOT THIS ENTRY, TRY NEXT ONE
0042                          ISUB=ISUB+1

```

```

PDP-11 FORTRAN-77 V4.0-1 16:58:13 15-Jul-86 Page 19
ACJAGC.FTN.36 /F77/TR:BLOCKS/MR

0043 IF(I$UB.NE.J$UB) THEN
0044   GOTO 20
0045 C HASH TABLE OVERFLOW, MUST COMPUTE, CAN NOT STORE
0046   X=F1(F(BETA))
0047   END IF
0048 C ELSE IF(I.T.EQ.BETA .AND. IT.EQ.I+1) THEN
0049   GOT IT!
0050   X=F1V(I$UB)
0051   END IF
0052   PROD=PROD*D1(X,NCHAN(I))
0053   CONTINUE
0054   I$UB=IHASH(BETA,2053)
0055   JSUB=I$UB
0056   T=PUI1(I$UB)
0057   IT=PUI1(I$UB)
0058 C IF(I.T.EQ.0.DO .AND. IT.EQ.0) THEN
0059   NOT FOUND, COMPUTE IT AND ENTER INTO TABLE
0060   Y=PUI1(BETA)
0061   PUIV(I$UB)=Y
0062   IPUIT(I$UB)=NJ+2
0063   ELSE IF(T.NE.BETA .OR. IT.NE.NJ+2) THEN
0064   NOT THIS ENTRY, TRY NEXT
0065   ISUB=ISUB+1
0066   IF(I$UB.GT.2053) ISUB=ISUB-2053
0067   IF(I$UB.NE.J$UB) THEN
0068   ELSE
0069   C HASH TABLE OVERFLOWED
0070   END IF
0071 C ELSE IF(I.T.EQ.BETA .AND. IT.EQ.NJ+2) THEN
0072   GOT IT!
0073   Y=PUIV(I$UB)
0074   END IF
0075   PRGRD=Y*(1.00-PR00)
0076   RETURN
END

F-11

```

```

POP-11 FORTRAN-77 V4.0-1 16:58:22 15-Jul-86 Page 20
ACJAGC.FTN.36 /F77/TR:BLOCKS/MR

0031 DOUBLE PRECISION FUNCTION PUI1(Y)
0032 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0033 DIMENSION WORK(75),STACK(75),SAVE(75)
0034 EXTERNAL DGXVI,PUIG
0035 C COMMON /PU1COM/ YY
0036 C AJ = NONCENTRAL PARAMETER FOR JAMMED HOPS
0037 C ANJ = NONCENTRAL PARAMETER FOR NON-JAMMED HOPS
0038 C NJ = NUMBER OF JAMMED HOPS
0039 C NJJ = NUMBER OF NON-JAMMED HOPS
0040 C COMMON /PU1PAR/ AJ,ANJ,NJ,NJJ
0041 C COMMON /PARMS/ BIGK,LL
0042 C IF(Y.LE.0.00) THEN
0043   PUI1=0.00
0044   RETURN
0045 C END IF
0046 C C NJ HAS ALREADY HAD 1 SUBTRACTED FOR RAPID CHI-SQUARE EVALUATION...
0047 C IF(NJ.NE.-1 .AND. NJ.NE.LL-1) THEN
0048 C WE MUST CONVOLVE TWO NONCENTRAL CHI-SQUARE DENSITIES
0049 C YY=Y
0050 C CALL ADQUA2(0,Y,VALUE,DXVI,PUI1,1,D-10,WORK,STACK,
0051 C           SAVE,75,KODE)
0052 C IF(KODE.NE.0) THEN
0053   WRITE(5,1) KODE
0054   1 FORMAT(' PUI1NVA2 ERROR: KODE='',12')
0055   STOP 'FATAL ERROR'
0056   END IF
0057   PUI1=VALUE/BIGK
0058 C WE ONLY HAVE ONE NONCENTRAL CHI-SQUARE DENSITY WITH 2*LL
0059 C DEGREES OF FREEDOM
0060 C IF(NJ.EQ.-1) THEN
0061   ALL HOPS UNJAMMED
0062   CALL CHISQE(Y,L1-1,ANJ,F,KODE)
0063   ELSE IF(NJ.EQ.LL-1) THEN
0064   ALL HOPS JAMMED
0065   CALL CHISQE(Y/BIGK,LL-1,AJ,F,KODE)
0066   F=F/BIGK
0067   END IF
0068 C IF(KODE.NE.0) THEN
0069   WRITE(5,1) KODE
0070   1 FORMAT(' BESSSEL FUNCTION ERROR CODE = ',11)
0071   STOP 'FATAL ERROR'
0072   END IF
0073   PUI1=F
0074   END IF
0075   RETURN
END

```

```

PDP-11 FORTRAN-77 V4.0-1 16:58:24 15-Jul-86
ACJAGC.FTN;36 /F77/TR:BLOCKS/NR

0001 DOUBLE PRECISION FUNCTION PU16(X)
0002 IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003 COMMON /PU16/ Y
          AJ = NONCENTRAL PARAMETER FOR JAMMED HOPS
          AJ = NONCENTRAL PARAMETER FOR NON-JAMMED HOPS
          AJ = NUMBER OF JAMMED HOPS
          MNJ = NUMBER OF NON-JAMMED HOPS
          COMMON /PU16PAR/ AJ, MNJ, MNJ
          COMMON /PARMS/ BIGK, LL
          CALL CHISQE(YX/BIGK, MNJ, AJ, F1, KODE)
0004 IF(KODE .NE. 0) THEN
0005 WRITE(5,1) KODE
0006 FORMAT(1X,'BESEL FUNCTION ERROR CODE: ',I2)
0007 STOP 'FATAL IN JAMMED HOP DENSITY'
0008 END IF
0009 CALL CHISQE(YY-X, MNJ, AJ, F2, KODE)
0010 IF(KODE .NE. 0) THEN
0011 WRITE(5,1) KODE
0012 STOP 'FATAL IN UNJAMMED HOP DENSITY'
0013 END IF
0014 PU16=F1+F2
0015 RETURN
0016 END
0017 END IF
0018 END IF
0019

```

Page 21

```

PDP-11 FORTRAN-77 V4.0-1 16:58:27 15-Jul-86
ACJAGC.FTN;36 /F77/TR:BLOCKS/NR

0001 DOUBLE PRECISION FUNCTION FL(ALPHA,L)
0002 IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003 COMMON /PARMS/ BIGK, LL
0004 IF(LL .NE. 0.AND.L.NE.LL) THEN
0005     R=(BIGK-L)/BIGK
0006     BIGLOG=DLG(BIGK)
0007     ARG=L*BIGLOG+ALPHA/2.
0008     START=DEXP(-ARG)
0009     FL=L-1
0010     DO 100 N=0,100
0011     EN=N
0012     IF(N.EQ.0) THEN
0013     PART=1.00
0014     TERM=DEXP(ALPHA/2.00,START,LL-1)
0015     SUM+=TERM
0016     PART=PART*R*(EN+FL)/EN
0017     TERM=PART*DEXP(ALPHA/2.00,START,LL+N-1)
0018     DUMMY=SUM+TERM
0019     SUM+=DUMMY
0020     END IF
0021     IF(DABS(TERM).LE.1.D-11*DABS(SUM)) GOTO 125
0022     CONTINUE
0023     STOP 'FL SUM DID NOT CONVERGE'
0024 ELSE
0025     IF(L.EQ.0) THEN
0026         A=ALPHA/2.00
0027     ELSE IF(LL.EQ.LL) THEN
0028         A=ALPHA/(2.D0*BIGK)
0029     END IF
0030     START=DEXP(-A)
0031     SUM=DEXP(A,START,LL-1)
0032     END IF
0033     FL=1.D0-SUM
0034     RETURN
0035 END
0036

```

Page 22

PDP-11 FORTRAN-77 V4.0-1 16:58:30 15-Jul-86 Page 23
ACJAGC.FTN;36 /F77/TR:BLOCKS/MR

```

0001 DOUBLE PRECISION FUNCTION DEXP(X,START,1UP)
0002 IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0003 TERM=START
0004 DEXP=TERM
0005 IF(1UP.EQ.0) RETURN
0006 DO 100 I=1,1UP
0007 F=I
0008 TERM=TERM*X/F
0009 DEXP=DEXP+TERM
0010 CONTINUE
0011 RETURN
0012 END
100

```

POP-11 FORTRAN-77 V4.0-1 16:58:32 15-Jul-86 Page 24
ACJAGC.FTN;36 /F77/TR:BLOCKS/MR

```

0001      C SUBROUTINE CHISQE(X,N,A,DEN,KODE)
0002      C   NON-CENTRAL CHI-SQUARE DENSITY FOR EVEN DEGREES OF FREEDOM
0003      C   DEGREES OF FREEDOM (N) IS N=2*N+2
0004      C
0005      C IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0006      C B=DSORT(X*A)
0007      C CALL DBESI(B,N,BESSEL,KODE)
0008      C IF(KODE.NE.0) RETURN
0009      C R=X/A
0010      C IF(R.NE.0.00) THEN
0011      C   POWER=R**2/(N/2.00)
0012      C ELSE
0013      C   POWER=1.00
0014      C END IF
0015      C DEN=0.5DD0*POWER*DEXP(B-0.5DD0*(X+A))+BESSEL
0016      C RETURN
0017      C
0018      C

```

PDP-11 FORTRAN-77 V4.0-1 16:58:33 15-Jul-86
ACJAGC.FTN;36 /F77/TR:BLOCKS/MR

Page 25

PDP-11 FORTRAN-77 V4.0-1 16:58:33 15-Jul-86
ACJAGC.FTN;36 /F77/TR:BLOCKS/MR

Page 26

0001 SUBROUTINE ADQUA2(XL,XU,Y,OR,F,TOL,WORK,STACK,SAVE,N,KODE)

C ADAPTIVE QUADRATURE ALGORITHM
C XL - LOWER LIMIT OF INTEGRAL (IN)
C XU - UPPER LIMIT OF INTEGRAL (IN)
C Y - VALUE OF INTEGRAL (OUT)
C OR - NAME OF A QUADRATURE RULE SUBROUTINE (IN)
C
C WITH CALLING SEQUENCE
C CALL QR(OR,XL,XU,F,Y)
C F - NAME OF FUNCTION TO BE INTEGRATED (IN)
C TOL - ERROR TOLERANCE FOR FINAL ANSWER (IN)
C WORK - WORK ARRAY OF SIZE N (IN)
C STACK - SECOND WORK ARRAY OF SIZE N, MUST NOT BE
C SAME AS WORK (IN)
C SAVE - THIRD WORK ARRAY OF SIZE N, MUST NOT BE
C SAME AS WORK NOR SAME AS STACK (IN)
C N - SIZE OF WORK AND STACK; MAX. NO. OF BISECTIONS (IN)
C KODE - ERROR INDICATOR (OUT)
C 0 -- NO ERROR
C 1 -- WORK ARRAYS TOO SMALL
C 2 -- EPS DIVIDED TO ZERO, EITHER ASKING FOR TOO
C TIGHT A TOLERANCE OR ROUND-OFF PREVENTS
C ATTAINING REQUIRED ACCURACY

C R. H. FRENCH, 14 AUGUST 1984

0002 IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003 EXTERNAL F
0004 DIMENSION WORK(N),STACK(N),SAVE(N)
0005 KODE=0
0006 Y=0.D0
0007 WORK(1)=XU
0008 A=XL
0009 NPTS=1
0010 EPS=TOL
0011 STACK(1)=EPS
0012 CALL QR(XL,XU,F,T)
0013 SAVE(1)=T
0014 10 B=MORD(NPTS)
0015 XM=(A+B)*0.5D0
0016 CALL QR(XM,F,P1)
0017 CALL QR(XM,B,F,P2)
0018 IF(DABS(T-P1-P2).LE.EPS) GOTO 20
0019 C SPLIT IT
0020 NPTS=NPTS+1
0021 IF(NPTS.GT.N) THEN
0022 KODE=i
0023 RETURN
0024 WORK(NPTS)=XM
0025 EPS=EPS/2.D0

F-14
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025

PDP-11 FORTRAN-77 V4.0-1 16:58:36 15-Jul-86
ACJAGC.FTN;36 /F77/TR:BLOCKS/MR

0001 FUNCTION IHASH(BETA,ISIZE)
0002 C AD HOC HASHING FUNCTION
0003 IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0004 IF(BETA.LT.1.0D) THEN
0005 IF(BETA.GT.0.5D0) THEN
0006 B=1.0D/(1.01D-BETA)
0007 ELSE
0008 B=50.0D+1.5D0/(BETA+0.01D0)
0009 END IF
0010 ELSE
0011 B=100000.0D*(BETA-DINT(BETA**1000.0D))/1000.0D)+187.00
0012 END IF
0013 B=B*23.00
0014 SIZE=ISIZE
0015 I=DMOD(B+0.5D0,SIZE)+0.5D0
0016 JHSH=I+1
0017 RETURN
END

Page 27

PDP-11 FORTRAN-77 V4.0-1 16:58:38 15-Jul-86
ACJAGC.FTN;36 /F77/TR:BLOCKS/MR

0001 SUBROUTINE D616(A,B,F,ANSWER)
0002 C 16-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL
0003 C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4
0004 C R. H. FRENCH, 28 FEBRUARY 1986
0005 C IMPLICIT DOUBLE PRECISION (A-H,0-Z)
0006 DIMENSION X(8),W(8)
0007 DATA X/ 0.095012509837637440185D0,
0008 \$ 0.281603550779258913230D0,
0009 \$ 0.458016777657227386342D0,
0010 \$ 0.617876244026437484470D0,
0011 \$ 0.75540408355003013895D0,
0012 \$ 0.865631202387831743860D0,
0013 \$ 0.94457502307323257607880D0,
0014 \$ 0.989400934991649932596D0/,
0015 \$ 0.189450610455068496285D0,
0016 \$ 0.182603415044923588867D0,
0017 \$ 0.169156519395002538189D0,
0018 \$ 0.149595988165776732881D0,
0019 \$ 0.124628911255533872052D0,
0020 \$ 0.095158511682492784810D0,
0021 \$ 0.062253523938647892863D0,
0022 \$ 0.027152459411754094852D0/,
0023 ANSWER=0.0D
0024 BHAD02=(B-A)/2.0D
0025 BPAD02=(B+A)/2.0D
0026 DO 10 I=1,8
0027 C=X(I)*BHAD02
0028 Y1=BPA02+C
0029 Y2=BPAD02-C
0030 ANSWER=ANSWER+W(I)*(F(Y1)+F(Y2))
0031 CONTINUE
0032 ANSWER=ANSWER*BHAD02
0033 RETURN
END

10

0015
0016
0017

Page 28

PDP-11 FORTRAN-77 V4.0-1 16:58:41 15-Jul-86
ACJAGC.FTN;36 /F77/TR-BLOCKS/MR

PDP-11 FORTRAN-77 V4.0-1 16:58:43 15-Jul-86
ACJAGC.FTN;36 /F77/TR-BLOCKS/MR

0001 SUBROUTINE DGXV1(A,B,F,ANSWER)

C 16-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL

C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4

C R. H. FRENCH, 28 FEBRUARY 1986

C

0002 IMPLICIT DOUBLE PRECISION (A-H,O-Z)

0003 DIMENSION X(8),W(8)

0004 DATA X/ 0.09501250983763744018500,

0.28160350779258691323000,

0.4580167765722738634200,

0.61787624440264374844700,

0.75540440835500301389500,

0.86563120238783174388000,

0.944575023073232576707800,

0.98940093499164993259600 /

DATA W/ 0.18945061045506849628500,

0.18260341504492358886700,

0.16915651939500253818900,

0.1495959881657673208100,

0.1246289712553387205200,

0.09515851168219278481000,

0.06225352393864789286300,

0.02715245941175409485200 /

ANSWER=0.00

BPA01=(B-A)/2.00

BPA02=(B+A)/2.00

DO 10 I=1,8

C=X(I)*BPA02

Y1=BPA02+C

Y2=BPA02-C

ANSWER=ANSWER+(1)*(F(Y1)+F(Y2))

CONTINUE

ANSWER=ANSWER+BPA02

RETURN

END

10

0005

0.0260853728600 /

DATA W/ 0.2395781703100,

0.566010084279300,

0.88700826291900,

1.2236644021500,

1.574448216300,

1.9647519765300,

2.315020566400,

2.770419268300,

3.2556433464000,

3.806311714200,

4.458477538400,

5.2700177844300,

6.3595334697300,

8.0317876327200,

11.527772100900 /

RESULT=0.00

0006 DO 10 J=1,15

RESULT=RESULT+W1)*F(A+X(J))

10 CONTINUE

0007 RETURN

0008 END

0009

0010

0011

```

PDP-11 FORTRAN-77 V4.0-1   16:58:46  15-Jul-86  Page 31
ACJAGC.FTR;36          /F77/TR:BLOCKS/WR

0001      C RANDOM MFS/FH IN PARTIAL BAND TONE JAMMING,
0001      C GIVEN A JAMMING EVENT
0001      C
0001      C JSUB - JAMMING EVENT VECTOR
0001      C LL - NUMBER OF HOPSYMBOLS
0001      C MM - ALPHABET SIZE
0001      C PROB - RESULTING CONDITIONAL ERROR PROBABILITY
0001      C
0002      C IMPLICIT DOUBLE PRECISION(A-H,D-Z)
0002      INTEGER R0,RL
0003      INTEGER JSUB(4)
0004      DIMENSION M(0:4),C0(0:12),CL(0:12),
0005      COEFK(3),COEFP(3),
0005      $           INDEX(9),INDUP(9),
0005      $           L
0006      C SYMBOLIC NAMES FOR INDEX-IDENTIFICATION SUBSCRIPTS
0006      C (LOOP NUMBER MODULO 3)
0006      C PARAMETER(LLOOR=1, LOOPX=2, LOOYP=0)
0007      COMMON /PARMS/ BIGK, LL
0008      COMMON /RRRCOM/ RHOT,RHON
0008      C
0009      C COMPUTE PARAMETERS
0009      C
0010      RHON=RHON1
0010      RHOT1=RHOT11
0011      LL=LL1
0012      BIGK=BIGK
0013      LM1=LL-1
0014      FKLL = D1(BIGK-1,DO,LM1)

0015      C COMPUTE POWERS OF FL
0015      DO 10 I=0,LL
0016      N(I)=0
0017      10  CONTINUE
0018      DO 12 I=2,MM
0019      N(JSUB(I))=N(JSUB(I))+1
0020      12  CONTINUE
0020      C OVER-ALL SUMMATION INITIALIZATION
0020      C
0021      SUM=0.00
0021      C
0022      C --- START LOOP ON R0 ---
0022      C
0023      DO 9000 R0=0,N(0)
0023      FRO=R0
0024      IF(R0.EQ.0) THEN
0025      COEFO=1.00
0026      ELSE
0027      COEFO=-COEFO*((N(0)-FRO+1.00)/FRO)
0028      END IF

```

```

PDP-11 FORTRAN-77 V4.0-1   16:58:46   15-Ju1-86
ACJAGC.F77/TR:BLOCKS/MR

Page 32

C --- START LOOP ON KO ---
C
C 0029 C KMAX=R0+LM1
C
C 0030 C PRE-COMPUTE THE J.C.P. MILLER COEFFICIENTS OVER KO!
C       CALL JCPMF(C0,KMAX,R0,LM1)
C       DO 8000 KO=0,KMAX
C
C 0031 C --- START LOOP ON RL ---
C
C 0032 C DO 7000 RL=0,M(L)
C 0033 C FRL=RL
C 0034 C IF(RL.EQ.0) THEN
C 0035 C COEFFL=1.00
C 0036 C ELSE
C 0037 C     COEFFL=COEFFL*((M(L))-FRL+1.00)/FRL
C 0038 C END IF

C --- START LOOP ON KL ---
C
C 0039 C KMAX=RL+LM1
C
C 0040 C PRE-COMPUTE THE J.C.P. MILLER COEFFICIENTS OVER KL!
C       CALL JCPMF(CL,KMAX,RL,LM1)
C       DO 6000 KL=0,KMAX
C       IF(KL.EQ.0) THEN
C 0041 C P0WKL=1.00
C 0042 C ELSE
C 0043 C     P0WKL=POWKL/BIGK
C 0044 C END IF
C 0045 C COEFFKL=CL(KL)*P0WKL
C 0046 C
C 0047 C --- START THE VARIABLE-LEVEL NESTED LOOPS
C
C 0048 C LOOP5=3*LM1
C 0049 C IF(LOOP5.EQ.0) GOTO 555
C
C 0050 C SET UP INDEX OF OUTERMOST LOOP
C 0051 C TINDEX(1)=0
C 0052 C SET NON-VARYING UPPER LIMITS
C 0053 C DO 22 I=1,LM1
C 0054 C     INDUP(3*I-2)=N(I)
C 22 CONTINUE
C
C 0055 C MARK OUTERMOST LOOP AS JUST ITERATED
C
C 0056 C NSUBB=1
C
C 0057 C PERFORM INITIALIZATION CODE FOR LOOPS FROM JUST-ITERATED LOOP
C INWARD

```

PDP-11 FORTRAN-77 V4.0-1 16:58:46 15-Jul-86

```

C FIRST. RESET INDICES OF ALL LOOPS WITHIN ONE JUST ITERATED
C
C 0055      DO 25 I=NSUB+1,LOOPS
C 0056          INDEX(1)=0
C 0057      CONTINUE
C
C SECOND. PERFORM FRONT-OF-LOOP CODE FOR EACH LOOP FROM ITERATE
C         LOOP ON INWARD
C
C
C 0058      DO 30 J=NSUB,LOOPS
C 0059          IRKP=MOD(J,3)
C 0060          MAP 1,2,3 INTO 1; 4,5,6 INTO 2; ETC. FOR NESTING LEVEL
C 0061          NEST=J+2/3
C 0062          IF (IRKP .EQ. 1)LOOP) THEN
C 0063              WE ARE ITERATING AN R-LOOP
C 0064              IF (INDEX(1) .EQ. 0) THEN
C 0065                  FIRST TIME THROUGH:
C 0066                      (A) INITIALIZE THE COEFFICIENT
C 0067                      COEF(NEST)=1.00
C 0068                  ELSE
C 0069                      NOT FIRST TIME THROUGH:
C 0070                      UPDATE THE COEFFICIENT
C 0071                      COEF(NEST)=-(COEF(NEST)/FRKLJ)*
C 0072                      ((INDEX(1)+1)-INDEX(1))
C 0073                  $ END IF
C 0074                  SET UP VARIABLE UPPER LIMITS FOR THE ASSOCIATED
C 0075                  K LOOP
C 0076                  INDP(1+1)=INDEX(1)
C 0077                  ELSE IF (IRKP .EQ. 1)LOOP) THEN
C 0078                  WE ARE ITERATING A K LOOP
C 0079                  IF (INDEX(1) .EQ. 0) THEN
C 0080                      FIRST TIME THROUGH:
C 0081                      INITIALIZE THE COEFFICIENT
C 0082                      COEF(NEST)=1.00
C 0083                  ELSE
C 0084                      NOT FIRST TIME THROUGH:
C 0085                      UPDATE THE COEFFICIENT
C 0086                      COEF(NEST)=COEF(NEST)*
C 0087                      ((INDEX(1)-1)-INDEX(1)+1.00)/INDEX(1)
C 0088                  $ END IF
C 0089                  SET UP THE UPPER LIMIT FOR THE ASSOCIATED P LOOP
C 0090                  INDP(1+1)=CAP(1,NEST,INDEX(1-1),INDEX(1))
C 0091                  ELSE IF (IRKP .EQ. 1)LOOP) THEN
C 0092                  WE ARE ITERATING A P LOOP
C 0093                  COEF(NEST)=DEFL(1,NEST,INDEX(1-2),INDEX(1-1),INDEX(1))
C 0094                  END IF
C 0095      CONTINUE
C
C     --- DO THE INTEGRAL
C

```

```

        DOUBLE PRECISION FUNCTION DEFILL(NEST,IR,K,IP,BIGK)
C THE FUNCTION D(P) FOR FBIN3/RMFSK
C
        IMPLICIT DOUBLE PRECISION(A-H,O-Z)
        DIMENSION G(0:8)
        IF(L1,L2).2 THEN
        IF(NEST.EQ.1) THEN
        DEF=DXI(BIGK,IR-K)
        IF(MOD(K,2).EQ.1) DEF=-DEF
        END IF
        ELSE IF(L1.EQ.3) THEN
        IF(NEST.EQ.1) THEN
        DEF=DXI(BIGK*BIGK,IR-K)*DBINCO(K,IP)*
        DXI(2.0D+BIGK-1.0D,K-1P)*DXI(BIGK-1.0D,IP)
        IF(MOD(K,2).EQ.1) DEF=-DEF
        ELSE IF(NEST.EQ.2) THEN
        DEF=DBINCO(IR-K,IP)*DXI(BIGK*(BIGK+2.0D),IR-K-1P)*
        DXI(BIGK-1.0D,IP)
        END IF
        ELSE IF(L1.EQ.4) THEN
        IF(NEST.EQ.1) THEN
        DEF=3.0D+BIGK*BIGK-3.0D*BIGK+1.0D
        DEF=(2.0D+BIGK+BIGK-3.0D*BIGK+1.0D)/DENOM
        G(0)=1.0D
        G(1)=K*COEF1
        BK1=BIGK-1.0D
        COEF2=0.5D+BK1*BK1/DENOM
        DO 41 N=2,IP
        41 6(N)=(K+1-N)*COEF1*G(N-1) + (2*K+2-N)*COEF2*G(N-2)/N
        CONTINUE
        DEF=DXI(BIGK,3*(IR-K))*DXI(DEMONM,K)*G(1P)
        IF(MOD(K,2).EQ.1) DEF=-DEF
        ELSE IF(NEST.EQ.2) THEN
        IP=4*IMOD(IP,K)
        ILDM=MAX(0,IP-IR+K)
        BK3=BIGK-3.0D
        TK1=3.0D*BIGK-1.0D
        BASE=BK3*BIGK/TK1
        SUMP=0.0D
        DO 42 10=1,IP
        42  SUMP=SUMP+DBINCO(IR-K,IP-1Q)*DBINCO(K,1Q)
        *DXI(BK3,IR-K-IP)*DXI(TK1,K)*DXI(BASE,1Q)
        CONTINUE
        DEF=SUMP*DXI(BIGK,IR-K-K-1P)*DXI(BIGK-1,0D,IP)
        ELSE IF(NEST.EQ.3) THEN
        43  G(0)=1.0D
        DEF=DXI(BIGK,3)-3.0D*RIGK*BIGK+3.0D*BIGK
        COEF1=(BIGK*BIGK-3.0D*BIGK+2.0D)/DENOM
        G(1)=(IR-K)*COEF1
        BK1=BIGK-1.0D
        COEF2=C.5D+BK1*BK1/(BIGK*DENO)
        44
    
```

PDP-11 FORTRAN-77 V4.0-1 16:59:08
ACJAGC.FTN;36 /F77/TR:BLOCKS/MR

0001 FUNCTION ICAPP(LL,NEST,IR,K)
C COMPUTE UPPER SUMMATION LIMIT PL
C

0002 IF(LL.LE.2) THEN
ICAPP=0
ELSE IF(LL.EQ.3) THEN
IF(NEST.EQ.1) THEN
ICAPP=K
ELSE IF(NEST.EQ.2) THEN
ICAPP=IR-K
END IF
ELSE IF(LL.EQ.4) THEN
IF(NEST.EQ.1) THEN
ICAPP=2*K
ELSE IF(NEST.EQ.2) THEN
ICAPP=IR
ELSE IF(NEST.EQ.3) THEN
ICAPP=2*(IR-K)
END IF
END IF
RETURN
END

0013
0014
0015
0016
0017
0018
0019
0020

Page 37

PDP-11 FORTRAN-77 V4.0-1 16:59:10
ACJAGC.FTN;36 /F77/TR:BLOCKS/MR

C THE INNER INTEGRAL FOR CONDITIONAL ERROR PROBABILITY

```

0001      C DOUBLE PRECISION FUNCTION GRAL(A0,IBO,LL)
0002      C
0003      C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0004      COMMON /RRDDM/ RHO1,RHO2
0005      COMMON /PARMS/ BIGK, LL
0006      A1=1.0D+AO
0007      IF(LL.GT.0.AND. LL.LT.LL) THEN
0008          BK1=BIGK-1.0D
0009          BASE=BK1/(1.0D+BIGK*AO)
0010          ARG1=(LL-LL)*RHO1/A1
0011          ARG2=(LL*RHO1*BIGK)/(1.0D+BIGK*AO)
0012          ARG3=ARG2*A1/BK1
0013          POWER=1.0D
0014          SUM=0.0D
0015          DO 100 I=0,IBO
0016              IF(IQ.GT.0) POWER=POWER*BASE
0017              PM=GMLSPM(IQ+LL-1,IBO-IQ,ARG1,PM)*GW_GPM(LL-1,IQ,ARG2,PM)+SUM
0018          CONTINUE
0019          F=1.0D
0020          DO 110 I=1,IBO
0021              F=F*(1/A1)
0022          CONTINUE
0023          F=(F/DX(A1,LL-1))/DX((1.0D+BIGK*AO,LL))
0024          GRAL=SUM+F*DEXP((ARG1+ARG2)*AO)
0025          ELSE IF(LL.EQ.0) THEN
0026              ARG1=-LL*RHO1/A1
0027              START=DEXP(ARG1*AO-LL*DLOG(A1))
0028              DO 200 I=1,IBO
0029                  START=START*(I/A1)
0030          CONTINUE
0031          GRAL=GMLSPM(LL-1,IBO,ARG1,START)
0032          ELSE IF(LL.EQ.LL) THEN
0033              A2=1.0D+BIGK*AO
0034              ARG1=-LL*RHO1/A2
0035              BASE=BIGK/A2
0036              START=DEXP(ARG1*BIGK*AO-LL*DLOG(A2))
0037              DO 300 I=1,IBO
0038                  START=START*(I*BASE)
0039          CONTINUE
0040          GRAL=GMLSPM(LL-1,IBO,ARG1,START)
0041      END IF
0042      RETURN
0043  END

```

POP-11 FORTRAN-77 V4.0-1 16:59:15 15-Jul-86 Page 39
 ACJAGC.FTN;36 /F77/TR:BLOCKS/MR

```

0001      DOUBLE PRECISION FUNCTION GRGLPM(IALFA,N,X,PREMUL)
C
C GENERALIZED LAGUERRE POLYNOMIALS WITH PREMULTIPLICATION BY
C A WEIGHTING FACTOR / TO DEFER THE INEVITABLE OVERFLOWS)
C
0002      IMPLICIT DOUBLE PRECISION(A-H,0-7)
0003      TERM=DBLIND(N+1,ALFA,N)*PREMUL
0004      SUM=TERM
0005      IF(N.EQ.0) GOTO 200
0006      DO 100 M=1,N
0007      TERM=TERM*((X/M)*((N-M+1.D0)/(1ALFA+M)))
0008      SUM=SUM+TERM
0009      CONTINUE
0010      100 200
0011      GRGLPM=SUM
0012      RETURN
0013      END
  
```

POP-11 FORTRAN-77 V4.0-1 16:59:17 15-Jul-86 Page 40
 ACJAGC.FTN;36 /F77/TR:BLOCKS/MR

```

0001      SUBROUTINE JCPMF(C,KMAX,R,LMAX)
C
C COMPUTE J.C.P. MILLER COEFFICIENTS DIVIDED BY K!
C
C IMPLICIT DOUBLE PRECISION(A-H,0-7)
C
0002      IMPLICIT DOUBLE PRECISION(A-H,0-7)
0003      INTEGER R
0004      DIMENSION C(0:KMAX)
0005      C(0)=1.00
0006      IF(LMAX.EQ.0) RETURN
0007      DO 100 K=1,KMAX
0008      SUM=0.D0
0009      MUP=DBLIND(K,LMAX)
0010      BC=1.D0
0011      DO 90 N=1,MUP
0012      BC=BC*((K-N+1.D0)/N)
0013      SUM=SUM+BC*((K+1.00)*N-K)*C(K-N)
0014      90  CONTINUE
0015      C(K)=SUM/K
0016      100  CONTINUE
0017      FAC=1.D0
0018      DO 110 K=1,KMAX
0019      FAC=FAC*K
0020      C(K)=C(K)/FAC
0021      110  CONTINUE
0022      RETURN
0023      END
  
```

PDP-11 FORTRAN-77 V4.0-1 16:59:20 15-Jul-86
 ACJAGC.FTN:36 /F77/TR:BLOCKS/SR

Page 41

```

0001      C SUBROUTINE MLINIT(LMAT,LL0W,LMAXC,LMAXR)
          C THIS SUBROUTINE INITIALIZES A "MATRIX DO-LOOP" STRUCTURE
          C DEFINED BY THE FOLLOWING PSEUDO-FORTRAN CODE:
          C DO 100 LMAT(1,1)=LL0W(1,1),LUP(1,1),LINC(1,1)
          C :
          C     DO 100 LMAT(LMAXC,1)=LL0W(LMAXC,1),LUP(LMAXC,1),LINC(LMAXC,1)
          C         LMAT(1,2)=LL0W(1,2),LUP(1,2),LINC(1,2)
          C :
          C     DO 100 LMAT(LMAXC,2)=LL0W(LMAXC,2),LUP(LMAXC,2),LINC(LMAXC,2)
          C         LMAT(1,2)=LL0W(1,2),LUP(1,2),LINC(1,2)
          C :
          C     DO 100 LMAT(1,LMAXR)=LL0W(1,LMAXR),LUP(1,LMAXR),LINC(1,LMAXR)
          C         LMAT(LMAXC,LMAXR)
          C :
          C     (STATEMENTS IN RANGE OF LOOP)
          C :
          C 100 CONTINUE
          C THE COMPANION ROUTINE MLITER HANDLES THE LOOP CONTROL AT THE
          C CONTINUE STATEMENT IN THE ABOVE STRUCTURE
          C USAGE:
          C     LOGICAL*1 GO
          C     DIMENSION LMAT(LMAXC,LMAXR),LL0W(LMAXC,LMAXR),LUP(LMAXC,LMAXR)
          C     DIMENSION LINC(LMAXC,LMAXR)
          C     (INITIALIZE MATRIX LMAT TO STARTING VALUES OF THE NESTED LOOPS)
          C     (INITIALIZE MATRIX LL0W TO STARTING VALUES OF THE NESTED LOOPS)
          C     (INITIALIZE MATRIX LUP TO INCREMENTS OF THE LOOPS)
          C     (INITIALIZE MATRIX LINC TO INCREMENTS OF THE LOOPS)
          C     CALL MLINIT(LMAT,LL0W,LMAXC,LMAXR)
          C 100 CONTINUE
          C     (STATEMENTS IN RANGE OF LOOPS)
          C     CALL MLITER(LMAT,LL0W,LUP,LINC,LMAXC,LMAXR,GO)
          C     IF(GO)GOTO 100
          WHERE
          LMAT = ARRAY FOR STORAGE OF LOOP INDICES. LMAT(1,1) IS THE
          OUTER-MOST LOOP. LMAT(LMAXC,LMAXR). THE INNER-MOST LOOP.
          LL0W = ARRAY FOR STORAGE OF LOOP STARTING VALUES. IN SAME
          SEQUENCE AS LMAT
          LUP = ARRAY FOR STORAGE OF LOOP FINAL VALUES. IN SAME
          SEQUENCE AS LMAT
          LINC = ARRAY FOR STORAGE OF LOOP INCREMENTS. IN SAME
          SEQUENCE AS LMAT
          LMAX = NUMBER OF LOOPS NESTED
          GO = LOGICAL VARIABLE. TRUE. IF JUMP BACK TO BEGINNING OF
          STATEMENTS IN THE RANGE OF THE LOOP SHOULD OCCUR.
          FALSE. OTHERWISE (I.E. OUTER-MOST LOOP TERMINATED)

```

PDP-11 FORTRAN-77 V4.0-1 16:59:20 15-Jul-86
 ACJAGC.FTN:36 /F77/TR:BLOCKS/SR

Page 42

```

          C PROGRAMMER: ROBERT H. FRENCH           DATE: 10 MARCH 1986
          C DIMENSION LMAT(LMAXC,LMAXR),LL0W(LMAXC,LMAXR)
          C DO 1 M=1,LMAXC
          C     DO 1 M=1,LMAXC
          C         LMAT(M,N)=LL0W(M,N)
          C     CONTINUE
          C     RETURN
          C END

          C PDP-11 FORTRAN-77 V4.0-1 16:59:22 15-Jul-86
          C ACJAGC.FTN:36 /F77/TR:BLOCKS/SR
          C
          C SUBROUTINE MLITER(LMAT,LL0W,LUP,LINC,LMAXC,LMAXR,GO)
          C
          C LOOP ITERATION LOGIC FOR A "MATRIX DO-LOOP"
          C SEE DETAILED COMMENTS IN SUBROUTINE MLINIT FOR USAGE AND
          C PARAMETER DEFINITIONS
          C
          C PROGRAMMER: ROBERT H. FRENCH           DATE: 10 MARCH 1986
          C
          C LOGICAL*1 GO
          C DIMENSION LMAT(LMAXC,LMAXR),LL0W(LMAXC,LMAXR),LUP(LMAXC,LMAXR)
          C DIMENSION LINC(LMAXC,LMAXR)
          C GO=.TRUE.
          C DO 100 MDX=1,LMAXR
          C     NSUB=LMAXR+1-MDX
          C     DO 100 MDX=1,LMAXC
          C         NSUB=LMAXC+1-MDX
          C         LMAT(NSUB,NSUB)=LMAT(1,1)+LINC(NSUB,NSUB)
          C         IF((LINC(NSUB,NSUB)=0.0).AND..LMAT(NSUB,NSUB).LE.LUP(NSUB))
          C             OR.
          C             (LINC(NSUB,NSUB).LT.0.AND..LMAT(NSUB,NSUB).GE.LUP(NSUB)))
          C     RETURN
          C     LMAT(NSUB,NSUB)=LL0W(NSUB,NSUB)
          C     CONTINUE
          C     GO=.FALSE.
          C     RETURN
          C END

          C PDP-11 FORTRAN-77 V4.0-1 16:59:22 15-Jul-86
          C ACJAGC.FTN:36 /F77/TR:BLOCKS/SR
          C
          C SUBROUTINE MLINIT(LMAT,LL0W,LUP,LINC,LMAXC,LMAXR)
          C
          C PROGRAMMER: ROBERT H. FRENCH           DATE: 10 MARCH 1986
          C
          C DIMENSION LMAT(LMAXC,LMAXR),LL0W(LMAXC,LMAXR),LUP(LMAXC,LMAXR)
          C DIMENSION LINC(LMAXC,LMAXR)
          C GO=.TRUE.
          C DO 100 MDX=1,LMAXR
          C     NSUB=LMAXR+1-MDX
          C     DO 100 MDX=1,LMAXC
          C         NSUB=LMAXC+1-MDX
          C         LMAT(NSUB,NSUB)=0.0
          C     RETURN
          C END

          C PDP-11 FORTRAN-77 V4.0-1 16:59:22 15-Jul-86
          C ACJAGC.FTN:36 /F77/TR:BLOCKS/SR
          C
          C SUBROUTINE MLITER(LMAT,LL0W,LUP,LINC,LMAXC,LMAXR)
          C
          C PROGRAMMER: ROBERT H. FRENCH           DATE: 10 MARCH 1986
          C
          C DIMENSION LMAT(LMAXC,LMAXR),LL0W(LMAXC,LMAXR),LUP(LMAXC,LMAXR)
          C DIMENSION LINC(LMAXC,LMAXR)
          C GO=.TRUE.
          C DO 100 MDX=1,LMAXR
          C     NSUB=LMAXR+1-MDX
          C     DO 100 MDX=1,LMAXC
          C         NSUB=LMAXC+1-MDX
          C         LMAT(NSUB,NSUB)=0.0
          C     RETURN
          C END

```

J. S. LEE ASSOCIATES, INC.

APPENDIX G COMPUTER PROGRAM FOR PLOTTING GRAPHICAL RESULTS FOR THE ANY-CHANNEL-JAMMED ADAPTIVE GAIN CONTROL RECEIVER

The following pages contain the source code listing for the FORTRAN-77 program used to produce the plotted graphical results for the any-channel-jammed adaptive gain control receiver for FH/RMFSK. The program makes use of the Hewlett-Packard Industry Standard Plotting Package (ISPP) to drive an HP-7470A plotter.

With minor modifications in annotations and file names, this program will serve to plot results for all other receivers. For brevity, the other versions of the plotting program are not included in this report.

PDP-11 FORTRAN-77 V4.0-1 10:47:21 17-Jul-8
ACJAGCP.FTN:10 F77/IR:BLOCKS/MR

POP-11 FORTRAN-77 V4.0-1 10:47:21 17-Jul-86
ACAJCP.FTN.19 /F77/TR:BLOCKS/WR Page 2

PROGRAM RNHOPP
C THIS PROGRAM PLOTS THE ERROR PROBABILITY FOR RANDOM M-ARY
C FSK/FH WITH MULTIPLE HOPS/BIT IN THE PRESENCE OF PARTIAL-BAND
C NOISE JAMMING
C ANALYSIS: L. E. MILLER, R. H. FRENCH
C PROGRAM: R. H. FRENCH

```

C DRAW THE BOX
C
0021 CALL SLOW
0022 CALL PLOT1(25,6,0,-3)
0023 CALL LOGAXI(0,0,'PROBABILITY OF BIT ERROR',8,0,-5,.5)
      | LEN('PROBABILITY OF BIT ERROR')
0024 CALL AXIS10(0,'BIT ENERGY TO JAMMING DENSITY RATIO (dB)',5)
      | LEN('BIT ENERGY TO JAMMING DENSITY RATIO (dB)')
      | 5..270.,0,10.)
C COMPLETE THE BOX
0025 CALL AXIS18(0,-1,0.5,270.,0,10.)
0026 CALL LOGAXI(0,-5.,1,0.8,0,0,5,.5)
C ANNOTATE M, L, EPMG
0027 CALL SYMBOL(7.5,-2.5,0.14,'M',-270.,2)
      | CALL SYMBOL(7.5,-2.5,0.14,'L',-270.,2)
      | ATTEMP:=M
      | CALL NUMBER(.999, .999, 0.14, RTENP, 270, -1)
      | CALL SYMBOL(.999, .999, 0.14, RTENP, 270, -1)
      | CALL SYMBOL(.999, .999, 0.14, RTENP, 270, -1)

```

```

C PROGRAM RINHAPP
C THIS PROGRAM PLOTS THE ERROR PROBABILITY FOR RANDOM M-ARY
C FSK/FH WITH MULTIPLE HOPS/BIT IN THE PRESENCE OF PARTIAL-BAND
C NOISE JAMMING
C
C ANALYSIS: L. E. MILLER, R. H. FRENCH
C PROGRAM: R. H. FRENCH
C
C V 3.0.0 - PLOTS
C
C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C PARAMETER (MJ=126)
C PARAMETER (N12=N+2)
C REAL*4 PRLOG(MJ),DBSAR(MJ),RTEMP
C CHARACTER*13 FNAME
C COMMON /INPUTS/ PASSES PARAMETERS OF THE RUN
C COMMON /INPUTS/ DEBIN(15),LLIST(4),NSLOTS,
C                 GEMBL(31),K,MM
C COMMON /SIZE/ PASSES NUMBERS OF PARAMETERS
C
S

```

G-2

```

0031 RTEMP,LL
0032 CALL NUMBER(999, 999, 0.15, RTEMP, 270, -1)
0033 CALL SYMBOL(7, 25, -2, 5, 0.14, 'E', 270, 1)
0034 CALL SYMBOL(7, 215, 999, 0.09, 'D', 270, 1)
0035 CALL SYMBOL(7, 25, 999, 0.14, 'N', 270, 2)
0036 CALL SYMBOL(7, 25, 999, 0.09, 'O', 270, 1)
0037 CALL SYMBOL(7, 25, 999, 0.14, 'I', 270, 1)
0038 RTEMP=DEBMO(110)
0039 CALL NUMBER(999, 999, 0.14, RTEMP, 270, .6)
0040 CALL SYMBOL(999, 999, 0.14, 'DB', 270, .3)
0041 RTEMP=NMSLOTS
0042 CALL NUMBER(7, 0, -2, 5, 0.14, RTEMP, 270, -1)
0043 CALL SYMBOL(999, 999, 0.14, 'SLOTS', 270, 6)
0044 CALL SYMBOL(6, 75, -2, 5, 0.14, 'AC3-ASC', 270, .7)
0045 CALL PENTUP
0046 DO 700 IG=1,MG
0047 GAMMA=GMNST(IG)
0048 16OUT=GMMA=1000.00+0.500

C OPEN DATA FILE
C
C
0049 WRITE(FNAME,730) MM,LL,10OUT,16OUT
0050 730 FORMAT('A',J1,J1,12,2,14,4,'.DAT')
0051 OPEN(UNIT=4,FILE=FNAME,STATUS='OLD',FORM='UNFORMATTED',
     ERR=740)
     $ GOTO 750
0052 READ(4,741) FNAME
0053 WRITE(5,741) FNAME
0054 FORMAT(1X,A13)
0055 STOP 'FILE NOT FOUND'
0056 READ(4,741) MM,LL,DENOM,ISLOTS,GAMMA,
     $(DBSR(1N),IN=1,N),
     $(PRLG(1N),IN=1,N)
0057 IF(MAIN,NE,MM, OR, LLIN,NE,LL, OR, DEBMO,NE,DEBMO(10),
     $ OR, ISLOTS,NE,ISLOTS, OR, GAMMA,NE,GAMMA) THEN
     $ WRITE(5,741) FNAME
     $ STOP 'FILE CONTENTS ERROR'
0058 END IF
C INTERPOLATE TO EDGE OF THE GRAPH
C
0061 DO 790 I=1,MJ
0062 NPTS=1
0063 IF(PRLG(1).GE.-5.) GOTO 790
0064 DY=PRLG(1)-PRLG(1-1)
0065 DX=DBSR(1)-DBSR(1-1)
0066 PART=(-5.0)-PRLG(1-1)
0067 SLOPE=DX/DY
0068 DBSR(1)=DBSR(1-1)+SLOPE*PART
0069 PRLG(1)=-5.
0070 GOTO 791
0071 CONTINUE

```

```

PDP-11 FORTRAN-77 V4.0-1    10:47:21   17-Jul-86   Page 3
0072    791  PRLOG(NPTS+1)=5,
0073          PR06(NPTS+2)=0, 625
0074          DSJN(NPTS+1)=0
0075          DSJN(NPTS+2)=-10.

C PLOT THE DATA
C
0076      CALL LINE(PRLOG,DSJN,NPTS,1,0,0)
0077      CALL PENUP
0078      CLOSE(UNIT=4)
0079      700  CONTINUE

C CLOSE AND FLUSH IF DONE, ELSE NEW PAGE
C
0080      IF(IL.EQ.ML.AND.10.EQ.MD) THEN
0081          CALL WEPEN(0)
0082          CALL PLOT(0.,0.,999)
0083          ELSE
0084              CALL WEPLT
0085          END IF
0086          800  CONTINUE
0087          900  CONTINUE
0088          STOP 0
0089      END

```

```

PDP-11 FORTRAN-77 V4.0-1    10:47:34   17-Jul-86   Page 4
ACJAGCP.FTN;10   /F77/TR:BLOCKS/MR
0001      SUBROUTINE GET
C
C INTERACTIVE INPUT OF PARAMETERS FOR RUN
C
C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C CHARACTER9 FIELD,BLANK9
C DIMENSION DG(31),DSMR(5,4)
C COMMON /INPUTS/ DEBOL(5),LLIST(4),NSLOTS,
C                 GAMSLT(31),K,PPM
C COMMON /SIZE/ MO,ML,NG
C
0002      DATA DG / .001D0, .002D0, -005D0,
0003          .010D0, .020D0, .050D0,
0004          .100D0, .200D0, .50D0, 1.00, 210.00/
0005      DATA DSMR /13.3524700, 12.313300, 10.944300, 0.00, 0.00,
0006          10.60652200, 9.628400, 8.352800, 0.00, 0.00,
0007          9.09401D0, 8.169000, 6.971995D0, 0.00, 0.00,
0008          8.07835D0, 7.199600, 6.069546D0, 0.00, 0.00/
0009      DATA BLANK9/,/
0010      WRITE(5,33)
0011      FORMAT(1X,BITS/SYMBOL (K) I7:, :,$)
0012      READ(5,3)K
0013      IF(K.EQ.0)K=2
0014      NM=2**K
0015      1      FORMAT(1X, HOW MANY EB/MO? [1]: :,$)
0016      2      READ(5,3)MO
0017      3      FORMAT(1I2)
0018      IF(MO.EQ.0)MO=1
0019      DO 7 IN=1,MO
0020      IF(K.LE.4) THEN
0021          D=DSMR(IN,K)
0022      ELSE
0023          D=0.0
0024      END IF
0025      4      WRITE(5,$)IN,DO
0026      FORMAT(1X,EB/MO(.12,.) F.9,6,'.'): :,$)
0027      5      READ(5,6)FIELD
0028      6      FORMAT(19)
0029      IF(FIELD.EQ.BLANK9) THEN
0030          DEBOL(IN)=DO
0031      ELSE
0032          DECODE(19,61,FIELD)DEBOL(IN)
0033          FORMAT(1X,F9.6)
0034      END IF
0035      CONTINUE
0036      7      WRITE(5,16)
0037      15     FORMAT(1X, HOW MANY L7 [1]: :,$)
0038      16     READ(5,3)NL
0039      IF(NL.EQ.0)NL=1
0040      DO 21 IN=1,ML
0041          WRITE(5,19)IN
0042          18     FORMAT(1X,L(' ,11,') I4): :,$)
0043          19

```

PDP-11 FORTRAN-77 V4.0-1 10:47:34 17-Jul-86 Page 5
 ACJAGCP.FTN;10 /F77/TR:BLOCKS/MR

```

0044      READ(5,3)ILLIST(IN)
          IF(ILLIST(IN).EQ.0)ILLIST(IN)=4
0045      CONTINUE
0046      21      WRITE(5,23)
0047      22      FORMAT(' HOPPING SLOTS? [2400]: ',\$)
0048      23      READ(5,24)NSLOTS
0049      24      FORMAT(15)
0050      25      IF(NSLOTS.EQ.0)NSLOTS=2400
0051      26      WRITE(5,26)
0052      27      FORMAT(' HOW MANY GAMMA? [10]: ',\$)
0053      28      READ(5,3)NG
0054      29      IF(NG.EQ.0)NG=10
0055      30      DO 31 IN=1,NG
0056      31      WRITE(5,29)IN,DG(IN)
0057      28      FORMAT(5,29)
0058      29      FORMAT(' GAMMA( ,12,:) [,1PDB,1,:]: ',\$)
0059      30      READ(5,30)GAMLIST(IN)
0060      31      FORMAT(015.8)
0061      31      IF(GAMLIST(IN).EQ.0.00)GAMLIST(IN)=DG(IN)
0062      31      CONTINUE
0063      31      RETURN
0064      END

```

J. S. LEE ASSOCIATES, INC.

APPENDIX H COMPUTER PROGRAM FOR CLIPPER RECEIVER WITH M=2 AND L=2

The following pages contain the source code listing for the FORTRAN-77 program used for numerical computations of the performance of the clipper receiver for the case of M=2 and L=2 with the jamming fraction $\gamma=q/N$ as a parameter.

```

PAGE 1
0001 PROGRAM CLPRAN
C THIS PROGRAM COMPUTES THE ERROR PROBABILITY FOR RANDOM M-ARY
C FSK/FK WITH 2 HOPSBIT IN THE PRESENCE OF PARTIAL-BAND
C NOISE JAMMING BY NUMERICAL INTEGRATION FOR THE CLIPPER RECEIVER
C
C ANALYSIS: L. E. MILLER, R. H. FRENCH
C PROGRAM: R. H. FRENCH
C V 2.1.0 - COMPUTATIONS ONLY
C
C
0002 IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003 PARAMETER (LJ=51)
0004 CHARACTER*1 YES, NO, REPLY, BLANK
0005 CHARACTER*13 FNAME, GNAME
0006 LOGICAL DOTAU
0007 LOGICAL*1 GOOD
0008 REAL*4 PRDGE(LJ), DBSR(LJ)
0009 VIRTUAL A(LJ), JASUB(100), C(625), ICSUB(625)
0010 VIRTUAL D(625), DSUB(625), PRERR(625), PSUB(625)
0011 C COMMON /INPUTS/ PASSES PARAMETERS OF THE RUN
0012 C COMMON /INPUTS/ DEBOL(3), NSLOTS, GAMSLT(10), K, MM
0013 C COMMON /SIZE/ NO, MG
0014 C COMMON /DEPAR/ BIGK, AB, BAB, LJAM,
0015 C COMMON /PARDEN/ RHOM, RHOT, RHOT2, RHOM4, RHOT4,
0016 C DATA YES, NO, BLANK /'Y', 'N', '-'/
0017 C CALL ERSET('29', TRUE.., FALSE.., TRUE.., FALSE.., 15)
0018 C CALL GET(INJ, START, DRINC)
0019 C MORITY=0.500*MM/(MM-1.00)
0020 C DO 800 ID=1, NO
0021 C     DBSR=10. DO 100*(DEBNOL(10)/10.00)
0022 C     RHOM=4.*EBNO/2.00
0023 C     RHOM2=2.*DBSR*RHOM
0024 C     RHOM=4.*DBSR*RHOM
0025 C     RHOM=8.*DBSR*RHOM
0026 C     RHOM=RHOM/2.00
0027 C     100UT=DEBNOL(10)
0028 C     DO 700 IG=1, NO
0029 C         GAMMA=GAMSLT(IG)
0030 C         HQ=GAMMA*MM*NSLOTS+Q_500
C
C OPEN DATA FILE
0031 IOUT=6
0032 WRITE(FNAME,730) MM, IOUT, 160UT
0033 FORMAT('C',J1,I2,'12.2,14.4',-,'.DAT')
0034 WRITE(6,776) MM, DEBNOL(10), GAMMA
0035 FORMAT( M_ ,J2,5X,'L2',5X,'EB/NO',-,'FR,4.5X,'GAMMA',-,'.1'

```

```

PP-11 FORTRAN-77 VA.0-1          13:12:26   16-Jul-86
CLIP2DOP.FTM;14    /F77/WR

0036      WRITE(5,733) FNAME
0037      733      OPEN(UNIT=4,FILE=FILENAME,STATUS='OLD',FORM='UNFORMATTED',
0038           $      ERR=750)
C HAVE AN EXISTING FILE, READ TO SEE HOW FAR WE GOT BEFORE
C
0039      1300     READ(4) MAIN, EBDIN, NSLIM, GAMIN, TAU, TAU2, PEO
C
C WE HAVE READ A VALUE OF TAU, SO WE WON'T NEED TO RECOMPUTE
C IT UNTIL EITHER EB/M0, M1, OR LL CHANGES
C
0040      DOTAU=.FALSE.
0041      JJ=0
0042      JJ=JJ+1
0043      READ(4,END=742) DBSR(JJ), PRLOG(JJ)
0044      GO TO 740
0045      CLOSE(UNIT=4)
0046      WRITE(5,7420)
0047      FORMAT(' ', FIX-UP RUN? [N] ', $')
0048      READ(5,7421) REPLY
0049      FORMAT(1A1)
0050      IF (REPLY.EQ.'BLANK') REPLY='N'
0051      IF (REPLY.EQ.'NO')  GOTO 755
0052      WRITE(5,7422)
0053      FORMAT(' HOW MANY GOOD POINTS? ', $)
0054      READ(5,7423) JGOOD
0055      FORMAT(13)
0056      IF (JGOOD.GT.JJ) JGOOD=JJ
C CREATE FILE CONTAINING ONLY GOOD DATA IN PROGRESS FORMAT
OPEN(UNIT=4,FILE=FILENAME,STATUS='NEW',FORM='UNFORMATTED')
0057      WRITE(4) MM,DEBN0(10),NSLOTS,GAMMA,TAU,TAU2,PEOC
0058      DO 7424 1-J,JGOOD
0059      WRITE(4) DBSR(1), PRLOG(1)
0060      CONTINUE
0061      7424     CLOSE(UNIT=4)
0062      JJ=JGOOD+1
0063      GOTO 755
0064
C NO EXISTING FILE, THIS IS THE FIRST TIME:
C CREATE FILE HEADER RECORD
0065      750      JJ=1
C ... WE CAN'T READ TAU FROM A FILE, SO IF THE FIRST TIME
C THROUGH THE LOOP ON EB/M0 WE MUST COMPUTE IT. BUT IF
C EB/M0 HASN'T CHANGED, WE DON'T NEED TO RECOMPUTE IT SINCE
C THE THRESHOLD IS NOT A FUNCTION OF GAMMA NOR OF EB/M1.
C
0066      CALL SETTAU(MM,PEOC)
0067      DOTAU=.FALSE.
0068
0069      WRITE(5,1991) MM, TAU
0070      WRITE(6,1991) MM, TAU
0071      FORMAT(1, M= ,12, L=2
0072      OPT THRES = '1P015.8)

```

```

OPEN(UNIT=4, FILE='FNAME', STATUS='NEW', FORM='UNFORMATTED')
WRITE(4, MM, DEBML(10), MSLOTS, GAMMA, TAU, TAU2, PE00
CLOSE(UNIT=4)
      MM, JGOUT
      WRITE('NAME,735'), MM, JGOUT
      FORMAT('EV,11,2',14.4,'.DAT')
      OPEN(UNIT=3, FILE='GNAME', STATUS='OLD', FORM='UNFORMATTED',
           READONLY, ERR=770)
      $      WRITE(5, 3939)
      3939  FORMAT(' READING EVENT FILE')
      READ(3, D, IDSUB, MUSED, 6000
      CLOSE(UNIT=3)
      6000 777
      CONTINUE
      C MUST CREATE EVENT PROBABILITIES
      770  CONTINUE
      0084  WRITE(5, 3938)
      3938  FORMAT(' CREATING EVENT FILE')
      CALL GENPTE(MM, NO, MSLOTS, 6000, A, JASUB, C, JCSUB,
                  D, IDSUB, MUSED)
      $      OPEN(UNIT=3, FILE='GNAME', STATUS='NEW', FORM='UNFORMATTED')
      WRITE(3, D, IDSUB, MUSED, 6000
      CLOSE(UNIT=3)
      IF(.NOT. GOOD) GOTO 700
      DO 600 IJ=JJ, MJ
      WRITE(5, 601) IJ
      601  FORMAT(' IJ= ', I3)
      DEBNJ=START+(IJ-1)*DB INC
      DESJR(IJ)=DEBNJ
      R=10.00** (DEBNJ/10.00)
      RHOT=GAMMA*R*EBRO/(GAMMA*R+EBRO)
      RHOT=R*RHOTS/2.00
      RHOT2=RHOT*2.00
      RHOT4=A. D*RHOT
      RHOT8=RHOT*8
      RHOT=RHOT/2.00
      CALL PSUBE(MM, PESYM, D, IDSUB, MUSED, PRERR, IPSUB, PE00)
      PE=WORBIT*PESTM
      WRITE(6, 666) DBSAR(LU), PE
      FORMAT(LX,F7.3,SX,1P012.5)
      PRLOG(IJ)=DLG010(PE)
      OPEN(UNIT=4, FILE='FNAME', STATUS='OLD', ACCESS='APPEND',
           FORM='UNFORMATTED')
      $      WRITE(4, DBSAR(IJ), PRLOG(IJ))
      CLOSE(UNIT=4)
      CONTINUE
      OPEN(UNIT=4, FILE='FNAME', STATUS='NEW', FORM='UNFORMATTED')
      CLOSE(UNIT=4)
      WRITE(4, MM, 2, DEBML(10), MSLOTS, GAMMA, DBSJR, PRLOG
      CONTINUE
      700  CONTINUE
      800  CONTINUE
      900  CONTINUE
      STOP 0
      0119
      0120
      0121
      0122
      0123
      0124
      0125
      0126
      0127
      0128
      0129
      0130
      0131
      0132
      0133
      0134
      0135
      0136
      0137
      0138
      0139
      0140
      0141
      0142
      0143
      0144
      0145
      0146
      0147
      0148
      0149
      0150
      0151
      0152
      0153
      0154
      0155
      0156
      0157
      0158
      0159
      0160
      0161
      0162
      0163
      0164
      0165
      0166
      0167
      0168
      0169
      0170
      0171
      0172
      0173
      0174
      0175
      0176
      0177
      0178
      0179
      0180
      0181
      0182
      0183
      0184
      0185
      0186
      0187
      0188
      0189
      0190
      0191
      0192
      0193
      0194
      0195
      0196
      0197
      0198
      0199
      0200
      0201
      0202
      0203
      0204
      0205
      0206
      0207
      0208
      0209
      0210
      0211
      0212
      0213
      0214
      0215
      0216
      0217
      0218
      0219
      0220
      0221
      0222
      0223
      0224
      0225
      0226
      0227
      0228
      0229
      0230
      0231
      0232
      0233
      0234
      0235
      0236
      0237
      0238
      0239
      0240
      0241
      0242
      0243
      0244
      0245
      0246
      0247
      0248
      0249
      0250
      0251
      0252
      0253
      0254
      0255
      0256
      0257
      0258
      0259
      0260
      0261
      0262
      0263
      0264
      0265
      0266
      0267
      0268
      0269
      0270
      0271
      0272
      0273
      0274
      0275
      0276
      0277
      0278
      0279
      0280
      0281
      0282
      0283
      0284
      0285
      0286
      0287
      0288
      0289
      0290
      0291
      0292
      0293
      0294
      0295
      0296
      0297
      0298
      0299
      0300
      0301
      0302
      0303
      0304
      0305
      0306
      0307
      0308
      0309
      0310
      0311
      0312
      0313
      0314
      0315
      0316
      0317
      0318
      0319
      0320
      0321
      0322
      0323
      0324
      0325
      0326
      0327
      0328
      0329
      0330
      0331
      0332
      0333
      0334
      0335
      0336
      0337
      0338
      0339
      0340
      0341
      0342
      0343
      0344
      0345
      0346
      0347
      0348
      0349
      0350
      0351
      0352
      0353
      0354
      0355
      0356
      0357
      0358
      0359
      0360
      0361
      0362
      0363
      0364
      0365
      0366
      0367
      0368
      0369
      0370
      0371
      0372
      0373
      0374
      0375
      0376
      0377
      0378
      0379
      0380
      0381
      0382
      0383
      0384
      0385
      0386
      0387
      0388
      0389
      0390
      0391
      0392
      0393
      0394
      0395
      0396
      0397
      0398
      0399
      0400
      0401
      0402
      0403
      0404
      0405
      0406
      0407
      0408
      0409
      0410
      0411
      0412
      0413
      0414
      0415
      0416
      0417
      0418
      0419
      0420
      0421
      0422
      0423
      0424
      0425
      0426
      0427
      0428
      0429
      0430
      0431
      0432
      0433
      0434
      0435
      0436
      0437
      0438
      0439
      0440
      0441
      0442
      0443
      0444
      0445
      0446
      0447
      0448
      0449
      0450
      0451
      0452
      0453
      0454
      0455
      0456
      0457
      0458
      0459
      0460
      0461
      0462
      0463
      0464
      0465
      0466
      0467
      0468
      0469
      0470
      0471
      0472
      0473
      0474
      0475
      0476
      0477
      0478
      0479
      0480
      0481
      0482
      0483
      0484
      0485
      0486
      0487
      0488
      0489
      0490
      0491
      0492
      0493
      0494
      0495
      0496
      0497
      0498
      0499
      0500
      0501
      0502
      0503
      0504
      0505
      0506
      0507
      0508
      0509
      0510
      0511
      0512
      0513
      0514
      0515
      0516
      0517
      0518
      0519
      0520
      0521
      0522
      0523
      0524
      0525
      0526
      0527
      0528
      0529
      0530
      0531
      0532
      0533
      0534
      0535
      0536
      0537
      0538
      0539
      0540
      0541
      0542
      0543
      0544
      0545
      0546
      0547
      0548
      0549
      0550
      0551
      0552
      0553
      0554
      0555
      0556
      0557
      0558
      0559
      0560
      0561
      0562
      0563
      0564
      0565
      0566
      0567
      0568
      0569
      0570
      0571
      0572
      0573
      0574
      0575
      0576
      0577
      0578
      0579
      0580
      0581
      0582
      0583
      0584
      0585
      0586
      0587
      0588
      0589
      0590
      0591
      0592
      0593
      0594
      0595
      0596
      0597
      0598
      0599
      0600
      0601
      0602
      0603
      0604
      0605
      0606
      0607
      0608
      0609
      0610
      0611
      0612
      0613
      0614
      0615
      0616
      0617
      0618
      0619
      0620
      0621
      0622
      0623
      0624
      0625
      0626
      0627
      0628
      0629
      0630
      0631
      0632
      0633
      0634
      0635
      0636
      0637
      0638
      0639
      0640
      0641
      0642
      0643
      0644
      0645
      0646
      0647
      0648
      0649
      0650
      0651
      0652
      0653
      0654
      0655
      0656
      0657
      0658
      0659
      0660
      0661
      0662
      0663
      0664
      0665
      0666
      0667
      0668
      0669
      0670
      0671
      0672
      0673
      0674
      0675
      0676
      0677
      0678
      0679
      0680
      0681
      0682
      0683
      0684
      0685
      0686
      0687
      0688
      0689
      0690
      0691
      0692
      0693
      0694
      0695
      0696
      0697
      0698
      0699
      0700
      0701
      0702
      0703
      0704
      0705
      0706
      0707
      0708
      0709
      07010
      07011
      07012
      07013
      07014
      07015
      07016
      07017
      07018
      07019
      07020
      07021
      07022
      07023
      07024
      07025
      07026
      07027
      07028
      07029
      07030
      07031
      07032
      07033
      07034
      07035
      07036
      07037
      07038
      07039
      07040
      07041
      07042
      07043
      07044
      07045
      07046
      07047
      07048
      07049
      07050
      07051
      07052
      07053
      07054
      07055
      07056
      07057
      07058
      07059
      07060
      07061
      07062
      07063
      07064
      07065
      07066
      07067
      07068
      07069
      07070
      07071
      07072
      07073
      07074
      07075
      07076
      07077
      07078
      07079
      07080
      07081
      07082
      07083
      07084
      07085
      07086
      07087
      07088
      07089
      07090
      07091
      07092
      07093
      07094
      07095
      07096
      07097
      07098
      07099
      070100
      070101
      070102
      070103
      070104
      070105
      070106
      070107
      070108
      070109
      070110
      070111
      070112
      070113
      070114
      070115
      070116
      070117
      070118
      070119
      070120
      070121
      070122
      070123
      070124
      070125
      070126
      070127
      070128
      070129
      070130
      070131
      070132
      070133
      070134
      070135
      070136
      070137
      070138
      070139
      070140
      070141
      070142
      070143
      070144
      070145
      070146
      070147
      070148
      070149
      070150
      070151
      070152
      070153
      070154
      070155
      070156
      070157
      070158
      070159
      070160
      070161
      070162
      070163
      070164
      070165
      070166
      070167
      070168
      070169
      070170
      070171
      070172
      070173
      070174
      070175
      070176
      070177
      070178
      070179
      070180
      070181
      070182
      070183
      070184
      070185
      070186
      070187
      070188
      070189
      070190
      070191
      070192
      070193
      070194
      070195
      070196
      070197
      070198
      070199
      070200
      070201
      070202
      070203
      070204
      070205
      070206
      070207
      070208
      070209
      070210
      070211
      070212
      070213
      070214
      070215
      070216
      070217
      070218
      070219
      070220
      070221
      070222
      070223
      070224
      070225
      070226
      070227
      070228
      070229
      070230
      070231
      070232
      070233
      070234
      070235
      070236
      070237
      070238
      070239
      070240
      070241
      070242
      070243
      070244
      070245
      070246
      070247
      070248
      070249
      070250
      070251
      070252
      070253
      070254
      070255
      070256
      070257
      070258
      070259
      070260
      070261
      070262
      070263
      070264
      070265
      070266
      070267
      070268
      070269
      070270
      070271
      070272
      070273
      070274
      070275
      070276
      070277
      070278
      070279
      070280
      070281
      070282
      070283
      070284
      070285
      070286
      070287
      070288
      070289
      070290
      070291
      070292
      070293
      070294
      070295
      070296
      070297
      070298
      070299
      0702100
      0702101
      0702102
      0702103
      0702104
      0702105
      0702106
      0702107
      0702108
      0702109
      0702110
      0702111
      0702112
      0702113
      0702114
      0702115
      0702116
      0702117
      0702118
      0702119
      0702120
      0702121
      0702122
      0702123
      0702124
      0702125
      0702126
      0702127
      0702128
      0702129
      0702130
      0702131
      0702132
      0702133
      0702134
      0702135
      0702136
      0702137
      0702138
      0702139
      0702140
      0702141
      0702142
      0702143
      0702144
      0702145
      0702146
      0702147
      0702148
      0702149
      0702150
      0702151
      0702152
      0702153
      0702154
      0702155
      0702156
      0702157
      0702158
      0702159
      0702160
      0702161
      0702162
      0702163
      0702164
      0702165
      0702166
      0702167
      0702168
      0702169
      0702170
      0702171
      0702172
      0702173
      0702174
      0702175
      0702176
      0702177
      0702178
      0702179
      0702180
      0702181
      0702182
      0702183
      0702184
      0702185
      0702186
      0702187
      0702188
      0702189
      0702190
      0702191
      0702192
      0702193
      0702194
      0702195
      0702196
      0702197
      0702198
      0702199
      0702200
      0702201
      0702202
      0702203
      0702204
      0702205
      0702206
      0702207
      0702208
      0702209
      0702210
      0702211
      0702212
      0702213
      0702214
      0702215
      0702216
      0702217
      0702218
      0702219
      0702220
      0702221
      0702222
      0702223
      0702224
      0702225
      0702226
      0702227
      0702228
      0702229
      0702230
      0702231
      0702232
      0702233
      0702234
      0702235
      0702236
      0702237
      0702238
      0702239
      0702240
      0702241
      0702242
      0702243
      0702244
      0702245
      0702246
      0702247
      0702248
      0702249
      0702250
      0702251
      0702252
      0702253
      0702254
      0702255
      0702256
      0702257
      0702258
      0702259
      0702260
      0702261
      0702262
      0702263
      0702264
      0702265
      0702266
      0702267
      0702268
      0702269
      0702270
      0702271
      0702272
      0702273
      0702274
      0702275
      0702276
      0702277
      0702278
      0702279
      0702280
      0702281
      0702282
      0702283
      0702284
      0702285
      0702286
      0702287
      0702288
      0702289
      0702290
      0702291
      0702292
      0702293
      0702294
      0702295
      0702296
      0702297
      0702298
      0702299
      0702300
      0702301
      0702302
      0702303
      0702304
      0702305
      0702306
      0702307
      0702308
      0702309
      0702310
      0702311
      0702312
      0702313
      0702314
      0702315
      0702316
      0702317
      0702318
      0702319
      0702320
      0702321
      0702322
      0702323
      0702324
      0702325
      0702326
      0702327
      0702328
      0702329
      0702330
      0702331
      0702332
      0702333
      0702334
      0702335
      0702336
      0702337
      0702338
      0702339
      0702340
      0702341
      0702342
      0702343
      0702344
      0702345
      0702346
      0702347
      0702348
      0702349
      0702350
      0702351
      0702352
      0702353
      0702354
      0702355
      0702356
      0702357
      0702358
      0702359
      0702360
      0702361
      0702362
      0702363
      0702364
      0702365
      0702366
      0702367
      0702368
      0702369
      0702370
      0702371
      0702372
      0702373
      0702374
      0702375
      0702376
      0702377
      0702378
      0702379
      0702380
      0702381
      0702382
      0702383
      0702384
      0702385
      0702386
      0702387
      0702388
      0702389
      0702390
      0702391
      0702392
      0702393
      0702394
      0702395
      0702396
      0702397
      0702398
      0702399
      0702400
      0702401
      0702402
      0702403
      0702404
      0702405
      0702406
      0702407
      0702408
      0702409
      0702410
      0702411
      0702412
      0702413
      0702414
      0702415
      0702416
      0702417
      0702418
      0702419
      0702420
      0702421
      0702422
      0702423
      0702424
      0702425
      0702426
      0702427
      0702428
      0702429
      0702430
      0702431
      0702432
      0702433
      0702434
      0702435
      0702436
      0702437
      0702438
      0702439
      0702440
      0702441
      0702442
      0702443
      0702444
      0702445
      0702446
      0702447
      0702448
      0702449
      0702450
      0702451
      0702452
      0702453
      0702454
      0702455
      0702456
      0702457
      0702458
      0702459
      0702460
      0702461
      0702462
      0702463
      0702464
      0702465
      0702466
      0702467
      0702468
      0702469
      0702470
      0702471
      0702472
      0702473
      0702474
      0702475
      0702476
      
```

PDP-11 FORTRAN-77 V4.0-1 13:12:43 16-Jul-86
 CLIP2HOP.FTN;14 /F77/MK

```

0047    IF(MJ.EQ.0) MJ=51
0048    IF(MJ.LT.0 .OR. MJ.GT.51) GOTO 32
0049    40      WRITE(5,41)
0050    41      FORMAT(' STARTING VALUE FOR EB/MJ (DB) [0.1: .5]')
0051    42      READ(5,A2,ERR=40) START
0052    42      FORMAT(F6.3)
0053    43      IF(MJ.EQ.1) RETURN
0054    35      WRITE(5,36)
0055    36      FORMAT(' DB INCREMENT FOR EB/MJ [1.0]: .5')
0056    37      READ(5,37,ERR=35) DBINC
0057    37      DBINC=1.00
0058    38      IF(DBINC.EQ.0.0D0) DBINC=1.00
0059    39      RETURN
0060

```

Page 5

PDP-11 FORTRAN-77 V4.0-1 13:12:50 16-Jul-86
 CLIP2HOP.FTN;14 /F77/MK

```

0001    SUBROUTINE PSUBEIM(P,D,JDSUB,MUSED,PRERR,JSUB,PE00)
0002    C COMPUTE UNCONDITIONAL ERROR PROBABILITY
0003    IMPLICIT DOUBLE PRECISION(A-H,D-Z)
0004    INTEGER JAM(4),LUP(4),JSUB(4)
0005    LOGICAL*1 GO,MONIE,STORE
0006    VIRTUAL PRERR(625),JSUB(625)
0007    VIRTUAL D1(625),JSUB(625)
0008    COMMON /SHARE2/ LO(4),LINC(4)
0009    $ COMMON /DEPAR/ BIGR, AAB, BAD, LJA,
0010    $ TAU, TAU2, TALK, TALK2
0011    DATA STORE/.TRUE./
0012    PE=0.00
0013    MPS=0
0014    DO 10 I=1,M
0015    LUP(I)=2
0016    JSUB(I)=0
0017    10  CONTINUE
0018    CALL LOC(M,LW,LUP,JSUB,ISUB)
0019    CALL PUTIN(PE00,PRERR,MPS,625,ISUB,KODE,STORE)
0020    IF(KODE.NE.0) STOP 'PRERR FULL'.
0021    JAM1=-1
0022    100   CONTINUE
0023    IF(JAM1.NE.JAM(1)) THEN
0024    JAM1=JAM(1)
0025    END IF
0026    CALL EVENT(M,JAM,PIE,D,JDSub,MUSED)
0027    C BYPASS CONDITIONAL ERROR PROBABILITY CALCULATION IF EVENT
0028    C PROBABILITY IS ZERO. THIS SAVES MUCH TIME.
0029    C SINCE JAMMING PROBABILITIES DEPEND ONLY ON NO. OF CHANNELS
0030    C JAMMED AND NOT THE ARRANGEMENT OF THE CHANNELS, WE CAN SORT
0031    C THE NON-SIGNAL CHANNELS INTO ASCENDING NUMBERS OF HOPS JAMMED.
0032    DO 111 I=1,M
0033    JSUB(I)=JAM(1)
0034    111   CONTINUE
0035    IF(JSUB(I).LT.JSUB(I+1)) THEN
0036    JTEMP=JSUB(I)
0037    JSUB(I)=JSUB(J)
0038    JSUB(J)=JTEMP
0039    END IF
0040    CONTINUE
0041    120   CONTINUE
0042

```

C STORE AS ELEMENT OF A SPARSE ARRAY TO SAVE MEMORY
C EVEN THOUGH WE STORE ZEROS. THE SORTING OF SUBSCRIPTS
C CUTS OUT MANY ELEMENTS.

0042 C TRY TO FIND CONDITIONAL ERROR PROBABILITY IN STORED ARRAY
CALL LOCNIM(L0M,LUP,JSUB) !
0043 IF(NOME) THEN
CALL LUPUP(PROB,PIERR,IPSUB,MPS,625,ISUB,STORE,NOME)
0044 C ... AND SAVE IT FOR POSSIBLE FUTURE RE-USE
CALL PSEL(JSUB,M,PROB)
0045 C SUM UP UNCONDITIONAL ERROR PROBABILITY
IF(KODE,NE,0) STOP 2
0046 PE=PE+PIE*PROB
0047 C ITERATE THE VECTOR-INDEX LOOP
101 CALL VLITER(JAM,LOW,LUP,LINC,M,GO)
IF(GO) GOTO 100
RETURN
0052
0053 END

PDP-11 FORTRAN-77 V4.0-1 13:12:56 /F77/MR
CLIP2HOP.FTN;14 /F77/MR

0001 SUBROUTINE EVENT(M,JAM,PIE,D,IDSUB,MUSED)

C COMPUTE LINEAR EQUIVALENT SUBSCRIPT FOR JAMMING EVENT
0002 DO 1 I=1,M
LUP(I)=2
0010 1 CONTINUE
C LOOK UP THE VALUE. GET 0.0 IF NOT THERE
CALL LOOKUP(PIE,D,IDSUB,MUSED,625,ISUB,STORE,NOME)
RETURN
0014

C SUBROUTINE GENIE(MQ,NSLOTS,GOOD,A,ISUB,C,ICSUB,
0,IDSUB,MUSED)

0001 \$
C SUBROUTINE TO GENERATE EVENT PROBABILITIES
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
LOGICAL*1 GO,GOZ,STORE,NOME,6000
DIMENSION LUP2(4),LUP3(4)
DIMENSION IUPA(4)
DIMENSION IUDP(4)
DIMENSION LUP1(4)
VIRTUAL AL(100),JASUB(100),C(625),ICSUB(625),
0002 0003 0004 0005 0006 0007 0008 0009 0010 0011
\$ DIMENSION I1(4),I11(4),I111(4)
C SHARED STORAGE FOR COMMON NEEDED CONSTANT ARRAYS
COMMON /SHARE2/L0M(4),LINC(4)
C SHARED STORAGE FOR: (1) INPUT DEFAULT LISTS,
C (2) CONDITIONAL PROB GEN., AND
C (3) EVENT PROB. GEN. THESE ARE NON-OVERLAPPING USAGES.
COMMON /SHARE/ISUB,ISUB2,AIN,L111,MM,AOUT,
ISUB,ISUB1,ISUB2,AIN,1,111,MM,AOUT,
CIN,COUT,DOUT,DIN
0012 0013 0014 0015 0016 0017 0018 0019 0020 0021 0022 0023 0024 0025 0026 0027 0028 0029 0030 0031 0032 0033
\$ DATA T100/100/
DATA TUPA/4*1/
DATA LUP1/4*1/
C STORE FALSE. => DON'T STORE ZERO ELEMENTS OF SPARSE ARRAY.
STORE=.FALSE.
GOOD=.TRUE.
IF(NOLE,0) THEN
GOOD=.FALSE.
RETURN
END IF
DO 80 L1=L1,MM
IUDP(L1)=2
80 CONTINUE
C JAMMING PATTERN W/MON-ZERO PROBABILITY ON PER-HOP BASIS
MUSA=A=0
C INITIALIZE VECTOR-INDEX LOOP
CALL VLINIT(L,LOW,MM)
90 CONTINUE
CALL LOCNIM(L0M,IOPA,JSUB)
CALL PRIMP(I,MM,NO,NSLOTS,AIN)
CALL PUTIM(AIN,A,ISUB,MUSA,1100,ISUB,IERR,STORE)
IF(IERR,NE,0)STCP 3
C ITERATE VECTOR-INDEX LOOP
CALL VLITER(L,LOW,LUP1,LINC,MM,GO)
IF(GO) GOTO 90
C COMPUTATION STARTS HERE. FIRST COPY A INTO D.
C SINCE ARRAYS ARE A(0:1,0:1,...:0:1) AND D(0:1,0:L,...:0:L)
C THE COPYING MUST BE DONE ON BASIS OF EQUIVALENT LINEAR
C SUBSCRIPTS RATHER THAN A SIMPLE MOVE OPERATION.
MUSED=0

```

POP-11 FORTRAN-77 V4.0-1   13:12:58  16-Jul-86
CLIPPHOP.FTN:14 /F77/NR

C INITIALIZE VECTOR-INDEX LOOP
CALL VLINIT(1,LOW,UP)
CONTINUE
CALL LOCM(MM,LOW,1UPA,I,ISUB1)
CALL LOCM(MM,LOW,1UPD,I,ISUB2)
CALL LOOKUP(AUDT,A,1ASUB,MUSEA,1100,ISU
CALL PUTUP(AUDT,D,1DSUB,MUSED,625,ISUB
C ITERATE VECTOR-INDEX LOOP
CALL VLITER(1,LOW,1UP1,LINC,MM,60)

```

Page 10

```

PDP-11 FORTRAN-77 V4.0-1 13:12:58 Page 10
CLIP2HOP.FTN:14 /F77/WR 16-Jul-86

C INITIALIZE VECTOR-INDEX LOOP
0034 CALL VINIT(1,L0W,MM)
0035 CONTINUE
0036 CALL LOCN(MM,LOW,IUPA,1,ISUB1)
0037 CALL LOCN(MM,LOW,IUPD,1,ISUB2)
0038 CALL LOOKUP(AOUT,A,IASUB,MUSEA,1100,ISUB1,STORE,NONE)
0039 CALL PUTIN(CIN,C,1CSUB,MUSE,MMAX,K,JERR,STORE)
0040 C ITERATE VECTOR-INDEX LOOP
0041 CALL VLITER(1,LOW,LUP1,LINC,MM,GO)
IF(GO1)GOTO 99
0042 C ... L-1 CONVOLUTIONS ARE NEEDED ...
0043 DO 9998 LL=1,1
0044 C SET UP VECTOR-LOOP UPPER LIMITS FOR THIS CONVOLUTION
0045 DO 125 MM=1,MM
0046 LUP2(MM)=L1
0047 LUP3(MM)=L1+1
0048 MUSEC=0
0049 CALL VINIT(1,LOW,MM)
0050 CONTINUE
0051 CALL VLIMIT(11,LOW,MM)
0052 CONTINUE
0053 DO 21 MM=1,MM
0054 111(MM)=1(MM)+11(MM)
0055 CONTINUE
0056 CALL LOCN(MM,LOW,IUPD,111,ISUB3)
0057 CALL LOOKUP(AOUT,A,IASUB,MUSEA,1100,ISUB1,STORE,NONE)
0058 CALL LOOKUP(DOUT,D,IDSUB,MUSED,625,ISUB2,STORE,NONE)
0059 CALL LOOKUP(COUT,C,ICSUB,MUSEC,625,ISUB3,STORE,NONE)
0060 CIN=COUT+AOUT*DOUT
0061 CALL PUTIN(CIN,C,1CSUB,MUSEC,625,ISUB3,1ERR,STORE)
0062 IF(JERR.NE.0) STOP 4
0063 CALL VLITER(11,LOW,LUP2,LINC,MM,602)
0064 IF(GO2)GOTO 97
0065 CALL VLITER(1,LOW,LUP1,LINC,MM,GO)
0066 IF(GO)GOTO 98
0067 MUSED=0
0068 CALL VLIMIT(11,LOW,MM)
0069 CONTINUE
0070 CALL LOCN(MM,LOW,IUPD,11,ISUB3)
0071 CALL LOOKUP(COUT,C,ICSUB,MUSEC,625,ISUB,STORE,NONE)
0072 DIN=DOUT
0073 CALL PUTIN(DIN,D,IDSUB,MUSED,625,1SUB,JERR,STORE)
0074 IF(JERR.NE.0) STOP 5
0075 CALL VLITER(11,LOW,LUP3,LINC,MM,GO)
0076 IF(GO)GOTO 96
0077 CONTINUE
0078 CALL LOCN(MM,LOW,IUPD,11,ISUB3)
0079 CALL LOOKUP(COUT,C,ICSUB,MUSEC,625,ISUB,STORE,NONE)
0080 DIN=DOUT
0081 CALL PUTIN(DIN,D,IDSUB,MUSED,625,1SUB,JERR,STORE)
0082 IF(JERR.NE.0) STOP 5
0083 CALL VLITER(11,LOW,LUP3,LINC,MM,GO)
0084 IF(GO)GOTO 96
0085 CONTINUE
0086 END

PDP-11 FORTRAN-77 V4.0-1 13:13:06 Page 11
CLIP2HOP.FTN:14 /F77/WR 16-Jul-86

0001 SUBROUTINE PUTIN(CIN,C,1CSUB,MUSE,MMAX,K,JERR,STORE)
0002 C THIS SUBROUTINE INSERTS AN ELEMENT INTO A SPARSE ARRAY FOR
0003 C WHICH ONLY THE NON-ZERO ELEMENTS ARE KEPT IN STORAGE IF
0004 C THE SWITCH STORE IS .TRUE.
0005 C THE DOUBLE PRECISION VALUE CIN IS STORED AS C(K) WHERE
0006 C THE AUXILIARY ARRAY ICSub IS USED TO KEEP TRACK OF THE
0007 C SUBSCRIPTS OF THE CORRESPONDING ENTRIES IN C.
0008 C USAGE:
0009 C   LOGICAL*1 STORE
0010 C   DOUBLE PRECISION C,CIN
0011 C   VIRTUAL ICSub(MAX),C(MAX)
0012 C   CALL PUTIN(CIN,C,1CSUB,MUSE,MMAX,K,JERR,STORE)
0013 C WHERE
0014 C   CIN = VALUE OF ELEMENT TO STORE
0015 C   C = ARRAY IN WHICH NON-ZERO VALUES ARE ACTUALLY STORED
0016 C   ICSub = AUXILIARY ARRAY FOR ACTUAL SUBSCRIPT VALUES
0017 C   MUSE = NUMBER OF ELEMENTS OF C CURRENTLY OCCUPIED
0018 C   MAX = SIZE OF ARRAY C
0019 C   JERR = ERROR RETURN CODE, 0 IF NO ERROR OR 1 IF THERE IS
0020 C   NO ROOM AVAILABLE IN C
0021 C   STORE = TRUE, TO STORE ZEROS EXPLICITLY, ELSE .FALSE.
0022 C   NOTE: IF CIN=0 AND THE SUBSCRIPT K IS FOUND IN ICSub, THEN
0023 C   THE ELEMENT IS DELETED BY SHIFTING DOWNWARD ALL
0024 C   FOLLOWING ELEMENTS OF THE ARRAY
0025 C PROGRAMMER: ROBERT H. FRENCH DATE: 11 JANUARY 1984
0026 IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0027 VIRTUAL ICSub(MAX),C(MAX)
0028 LOGICAL*1 STORE
0029 JERR=0
0030 IF(STORE) GOTO 5
0031 IF(CIN.EQ.0.0D0) GOTO 30
0032 DO 10 I=1,MUSE
0033 IF(ICSub(I).NE.K) GOTO 10
0034 C(I)=CIN
0035 RETURN
0036 CONTINUE
0037 IF(GO1)GOTO 20
0038 IF(MUSE.LT.MM) GOTO 20
0039 TERR=1
0040 RETURN
0041 MUSE=MUSE+1
0042 ICSub(MUSE)=K
0043 C(MUSE)=CIN
0044 RETURN
0045 END

```

PDF-11 FORTRAN-77 V4.0-1 /F77/WR
 CLIP2HOP.FTN;14

```

0021 30      DO 40 I=1,MUSE
0022      J=1
0023      IF(ICSUB(I).EQ.K) GOTO 50
0024      CONTINUE
0025      RETURN
C REMOVE THE ZEROED ELEMENT AND BUMP COUNT OF ENTRIES USED
C
0026 50      DO 60 I=J,MUSE-1
0027      ICSUB(I)=ICSUB(I+1)
0028      C(I)=C(I+1)
0029      CONTINUE
0030      MUSE=MUSE-1
0031      RETURN
END

```

Page 12

PDF-11 FORTRAN-77 V4.0-1 /F77/WR
 CLIP2HOP.FTN;14

Page 13

```

0001      SUBROUTINE LOOKUP(COUT,C,ICSUB,N,MMAX,K,STORE,NONE)
C THIS SUBROUTINE RETRIEVES AN ELEMENT OF A SPARSE ARRAY WHICH
C HAS BEEN STORED COMPACTLY BY STORING ONLY NON-ZERO ELEMENTS.
C THE ARRAY IS DOUBLE PRECISION.
C
C USAGE:
C   VIRTUAL ICSUB(MMAX), C(MMAX)
C   LOGICAL*1 STORE, NONE
C   DOUBLE PRECISION COUT
C   CALL LOOKUP(COUT,C,ICSUB,N,MMAX,K,STORE,NONE)
C WHERE
C   COUT = VALUE OF C(IK) (OUTPUT FROM SUBROUTINE)
C   C = ARRAY USED TO STORE NON-ZERO ELEMENTS
C   ICSUB = AUXILIARY ARRAY TO STOP - ACTUAL SUBSCRIPTS
C   N = NUMBER OF ELEMENTS C: C CURRENTLY IN USE
C   MMAX = SIZE OF C
C   K = SUBSCRIPT OF SPARSE ARRAY TO LOOK UP
C   STORE = .TRUE. IF ZEROES STORED EXPLICITLY, ELSE .FALSE.
C   NONE = .FALSE. IF ZEROES NOT STORED OR ZEROES STORED AND
C   ELEMENT IS FOUND IN THE STORED ARRAY
C   .TRUE. IF ZEROES ARE STORED AND THE ELEMENT IS
C   NOT FOUND (OUTPUT QUANTITY)
C
C PROGRAMMER: ROBERT H. FRENCH
C DATE: 11 JANUARY 1984
C
0002      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003      VIRTUAL ICSUB(MMAX),C(MMAX)
0004      LOGICAL*1 STORE, NONE
0005      NONE=.FALSE.
0006      DO 10 I=1,N
0007      IF(ICSUB(I)).NE.K)GOTO 10
0008      COUT=C(I)
0009      RETURN
10      CONTINUE
      IF(STORE) THEN
        NONE=.TRUE.
      ELSE
        COUT=0.
      END IF
      RETURN
END

```

0001 SUBROUTINE LOCN(NDIM,ILOW,IUP,ISUB,LINEAR)
 C THIS SUBROUTINE COMPUTES THE EQUIVALENT LINEAR SUBSCRIPT FOR
 C A MULTIDIMENSIONAL ARRAY OF NDIM DIMENSIONS
 C IF THE ARRAY A IS DEFINED AS
 C DIMENSION A(ILON(1):...,ILON(NDIM):IUP(NDIM))
 C AND ISUB(1),...,ISUB(NDIM) IS A SET OF SUBSCRIPTS FOR A,
 C THEN THIS SUBROUTINE RETURNS IN LINEAR THE OFFSET FROM THE
 C ORIGIN OF A TO THE ELEMENT A(ISUB(1),...,ISUB(NDIM)), ASSUMING
 C THE FIRST SUBSCRIPT VARIES MOST RAPIDLY.

USAGE:
 C DIMENSION ILON(NDIM),IUP(NDIM),ISUB(NDIM)
 C DATA ILON/Lower limits of defined subscripts of array/
 C DATA IUP/Upper limits of defined subscripts of array/
 C ...SET ISUB TO DESIRED SEQUENCE OF SUBSCRIPTS...
 C CALL LOCN(NDIM,ILON,IUP,ISUB,LINEAR)
 C WHERE
 C NDIM = NUMBER OF DIMENSIONS THE ARRAY HAS
 C ILON = ARRAY OF LOWER SUBSCRIPT BOUNDS
 C IUP = ARRAY OF UPPER SUBSCRIPT BOUNDS
 C ISUB = ARRAY CONTAINING SUBSCRIPT FOR WHICH LOCATION IS
 C TO BE COMPUTED
 C LINEAR = RETURNED VALUE OF OFFSET INTO ARRAY IN MEMORY

PROGRAMMER: ROBERT H. FRENCH
 DATE: 11 JANUARY 1984

0002 DIMENSION ILON(NDIM),IUP(NDIM),ISUB(NDIM)
 C LINEAR=0
 0003 DO 10 I=1,NDIM-1
 0004 J=NDIM-I+1
 0005 LINEAR=(ISUB(J)-ILON(J))*(IUP(J-1)-ILON(J-1)+1)
 0006 CONTINUE
 10 LINEAR=LINEAR+ISUB(1)-ILON(1)
 0007 RETURN
 END
 0008
 0009
 0010

0001 SUBROUTINE VLINIT(LVEC,LLON,LMAX)
 C THIS SUBROUTINE INITIALIZES A "VECTOR 00-LOOP" STRUCTURE
 C DEFINED BY THE FOLLOWING PSEUDO-FORTRAN CODE:
 C DO 100 LVEC(1)=LLON(1),IUP(1),LINC(1)
 C DO 100 LVEC(2)=LLON(2),IUP(2),LINC(2)
 C :
 C DO 100 LVEC(LMAX)=LLON(LMAX),IUP(LMAX),LINC(LMAX)
 C :
 C (STATEMENTS IN RANGE OF LOOP)
 C 100 CONTINUE
 C THE COMPANION ROUTINE VLITER HANDLES THE LOOP CONTROL AT THE
 C CONTINUE STATEMENT IN THE ABOVE STRUCTURE
 C USAGE:
 C LOGICAL *1 GO
 C DIMENSION LVEC(LMAX),LLON(LMAX),IUP(LMAX),LINC(LMAX)
 C (INITIALIZE ARRAY LLON TO STARTING VALUES OF THE NESTED LOOPS)
 C (INITIALIZE ARRAY IUP TO STOPPING VALUES OF THE NESTED LOOPS)
 C (INITIALIZE ARRAY LINC TO INCREMENTS OF THE LOOPS)
 C CALL VLINIT(LVEC,LLON,LMAX)
 C 100 CONTINUE
 C (STATEMENTS IN RANGE OF LOOP)
 C CALL VLITER(LVEC,LLON,IUP,LINC,LMAX,GO)
 C WHERE
 C LVEC = ARRAY FOR STORAGE OF LOOP INDICES. LVEC(1) IS THE
 C OUTER-MOST LOOP; LVEC(LMAX), THE INNER-MOST LOOP.
 C LLON = ARRAY FOR STORAGE OF LOOP STARTING VALUES, IN SAME
 C SEQUENCE AS LVEC
 C IUP = ARRAY FOR STORAGE OF LOOP FINAL VALUES, IN SAME
 C SEQUENCE AS LVEC
 C LINC = ARRAY FOR STORAGE OF LOOP INCREMENTS, IN SAME
 C SEQUENCE AS LVEC
 C MAX = NUMBER OF LOOPS NESTED
 C GO = LOGICAL VARIABLE, TRUE, IF JUMP BACK TO BEGINNING OF
 C STATEMENTS IN THE RANGE OF THE LOOP SHOULD OCCUR;
 C FALSE, OTHERWISE (I.E. OUTER-MOST LOOP TERMINATED)
 C PROGRAMMER: ROBERT H. FRENCH DATE: 11 JANUARY 1984
 C
 C DIMENSION LVEC(LMAX),LLON(LMAX)
 C DO 1 N=1,LMAX
 C LVEC(N)=LLON(N)
 C CONTINUE
 C RETURN
 C END

PDP-11 FORTRAN-77 V4.0-2 /F77/WR 13:13:16 16-Jul-86 Page 16
 CLIP2HOP.FTN;14 CLIP2HOP.FTN;14 /F77/WR

0001 SUBROUTINE VLITER(LVEC,LLDM,LUP,LINC,LMAX,G0)
 C LOOP ITERATION LOGIC FOR A "VECTOR DO-LOOP"
 C SEE DETAILED COMMENTS IN SUBROUTINE VLINIT FOR USAGE AND
 C PARAMETER DEFINITIONS

```

0002      C PROGRAMMER: ROBERT H. FRENCH
0003      C DATE: 11 JANUARY 1984
0004      LOGICAL*I1 GO
0005      DIMENSION LVEC(LMAX),LLDM(LMAX),LUP(LMAX),LINC(LMAX)
0006      DIMENSION I1(I1)
0007      KJAM=0
0008      CONTINUE
0009      IF(KJAM.GT.MIND(KQ,KM)) RETURN
0010      KPMAX=KJAM-1
0011      LPMAX=KM-KJAM-1
0012      JPMAX=KM-1
0013      PROD=1.D0
0014      Q=KQ
0015      DIFFRQ=KM-KQ
0016      EN=KN
0017      DO 100 LOOP=0,IMAX
0018      F=LLOOP
0019      IF(LOOP.LT.KPMAX) PROD=PROD*(0-F)
0020      IF(LOOP.LE.JPMAX) PROD=PROD/(EN-F)
0021      IF(LOOP.LE.LPMAX) PROD=PROD*(DIFFRQ-F)
0022      CONTINUE
0023      100 CONTINUE
0024      AIM=PROD
0025      RETURN
0026
  
```

```

0001      BLOCK DATA
0002      C INITIALIZE SHARED CONSTANTS
0003      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0004      COMMON /SHARE/ D(10),DSMR(3,4)
0005      COMMON /SHARE2/ LOM(4),LINC(4)
0006      C WEIGHTS AND ABSISSAS FOR 16-POINT GAUSSIAN QUADRATURE
0007      C DEFAULT LISTS FOR INTERACTIVE PARAMETER INPUTS
0008      C ARE SHARED WITH LARGE WORKING STORAGE ARRAYS SINCE THEY
0009      C MAY BE DESTROYED ONCE THE INPUT PARAMETERS ARE SET UP
0010      DATA DUG / .00100, .00200, .00500,
0011      C FREQUENTLY NEEDED CONSTANT ARRAYS AND SCALARS
0012      DATA LOM/4#0/,LINC/#1/
0013      END
0014      CONTINUE
0015      C SET UP VALUES WHICH WILL REMAIN IF THIS IS THE NOTHING-JAMMED CASE
0016      Q1=0.12*DSQRT(RHOMH).DSQRT(2.0*DUG)
0017      TAU=0.00
0018      C IF ANYTHING IS JAMMED, SET UP JAMMING-RELATED QUANTITIES
0019      IF (KSUB.NE.0) THEN
0020          B1GK=RHOM/RHOT
0021          AAB=B1GK/BK1
0022          BAB=1.00/BK1
0023          TALK=TAU/B1GK
0024          TALK2=TAU2/B1GK
0025          Q1=Q(2.0*DSQRT(RHOTH)).DSQRT(2.0*DUG)
0026      END IF
0027      COUNT NUMBER OF NONSIGNAL CHANNELS WITH LN HOPS JAMMED
0028      DO 10 1=0,2
0029      KCHAN(1)=0
0030      DO 11 1=2,MN
0031      KSUB=JSUB(1)
0032      KCHAN(KSUB)=KCHAN(KSUB)+1
0033      CONTINUE
0034      JSUB=JSUB(1)
0035      C DO THE CONTINUOUS PART OF THE DENSITY IN SECTIONS
0036      C

```

```

0035      PCOUNT=0.0D0
0036      DO 13 ISECT=1,2
0037      XL=(ISECT-1)*TAU
0038      XU=ISECT*TAU
0039      CALL ADQUAD(XL,XU,CHUNK,DG16,PERAND,1.0-8,WORK,
0040           STACK,HEAP,30,KODE)
0041      S    CALL TEST2(KODE,10)
0042      PCOUNT=PCOUNT+CHUNK
0043      13  CONTINUE
0044      C DO THE TIE PART OF THE DENSITY
0045      CALL TIES(JSUB,MM,PTIE)
0046      C PUT THEM TOGETHER
0047      C PROB=1.00-PCOUNT-PTIE
0048      RETURN
0049      END

PDP-11 FORTRAN-77 V4.0-1 /F77/MR
CLIPPHOP.FTN;14 /F77/MR
13:13:27  16-Jul-86
Page 21

```

```

0001      C ADAPTIVE QUADRATURE ALGORITHM
0002      C XL - LOWER LIMIT OF INTEGRAL (IN)
0003      C XU - UPPER LIMIT OF INTEGRAL (IN)
0004      C Y - VALUE OF INTEGRAL (OUT)
0005      C QR - NAME OF A QUADRATURE RULE SUBROUTINE (IN)
0006      C F - NAME OF FUNCTION TO BE INTEGRATED (IN)
0007      C TOL - ERROR TOLERANCE FOR FINAL ANSWER (IN)
0008      C WORK - WORK ARRAY OF SIZE N (IN)
0009      C STACK - SECOND WORK ARRAY OF SIZE N, MUST NOT BE
0010      C SAME ARRAY AS WORK (IN)
0011      C HEAP - THIRD WORK ARRAY OF SIZE N, DISTINCT FROM BOTH WORK AND STACK
0012      C N - SIZE OF WORK AND STACK: MAX. NO. OF BISECTIONS (IN)
0013      C KODE - ERROR INDICATOR (OUT)
0014      0 -- NO ERROR
0015      1 -- WORK ARRAYS TOO SMALL
0016      2 -- EPS DIVIDED TO ZERO, EITHER ASKING FOR TOO
0017          TIGHT A TOLERANCE OR ROUND-OFF PREVENTS
0018          ATTAINING REQUIRED ACCURACY
0019      C R. H. FRENCH, 14 AUGUST 1984

0020      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0021      EXTERNAL F
0022      DIMENSION WORK(N),STACK(N),HEAP(N)
0023      KODE=0
0024      Y=0.D0
0025      WORK(1)=XU
0026      HEAP(1)=T
0027      CALL QR(XL,XU,F,T)
0028      A=XL
0029      NPITS=1
0030      EPS=TOL
0031      STACK(1)=EPS
0032      B=WORK(NPITS)
0033      XH=(A+B)*0.5D0
0034      CALL QR(A,XH,F,P1)
0035      CALL QR(XH,B,F,P2)
0036      IF(DABS(P1-P2).LE.EPS) GOTO 20
0037      C SPLIT IT
0038      NPITS=NPITS+1
0039      IF(NPITS.GT.N) THEN
0040          KODE=1
0041          RETURN
0042      END IF
0043      WORK(NPITS)=XH
0044      HEAP(NPITS)=P2
0045      T=P1
0046      EPS=EPS/2.00

```

```

PDP-11 FORTRAN-77 V4.0-1 /F77/MR 13:13:30
CLIP2PROP.FTN;14 16-Jul-86

0028 IF(EPS.EQ.0.001) THEN
0029 KODE=2
0030 RETURN
0031 END IF
0032 STACK(NPTS)=EPS
0033 GOTO 10

C FINISHED A PIECE
20 Y=Y*P1+P2
EPS=STACK(NPTS)
T=HEAP(NPTS)
NPTS=NPTS-1
A=B
IF(NPTS.EQ.0) RETURN
GOTO 10
END

```

Page 23

POP-11 FORTRAN-77 V4.0-1 13:13:34 16-Jul-86
C:\P2\HPF.FTN\14 /F77/AB Page 24

```

0001      SUBROUTINE ADQDA2(XL,XU,Y,QR,F,TOL,WORK,STACK,HEAP,N,KODE)
0002
0003      C ADAPTIVE QUADRATURE ALGORITHM
0004      C XL - LOWER LIMIT OF INTEGRAL (IN)
0005      C XU - UPPER LIMIT OF INTEGRAL (IN)
0006      C Y - VALUE OF INTEGRAL (OUT)
0007      C QR - NAME OF A QUADRATURE RULE SUBROUTINE (IN)
0008      C WITH CALLING SEQUENCE
0009          CALL QR(XL,XU,F,T)
0010
0011      C F - NAME OF FUNCTION TO BE INTEGRATED (IN)
0012      C TOL - ERROR TOLERANCE FOR FINAL ANSWER (IN)
0013      C WORK - WORK ARRAY OF SIZE N (IN)
0014      C STACK - SECOND WORK ARRAY OF SIZE N. MUST NOT BE
0015      C SAME AS WORK (IN)
0016      C HEAP - THIRD WORK ARRAY OF SIZE N, DISTINCT FROM BOTH WORK AND S-
0017      C N - SIZE OF WORK AND STACK; MAX. NO. OF BISECTIONS (IN)
0018      C KODE - ERROR INDICATOR (OUT)
0019          0 -- NO ERROR
0020          1 -- WORK ARRAYS TOO SMALL
0021          2 -- EPS DIVIDED TO ZERO. EITHER ASKING FOR TOO
0022              TIGHT A TOLERANCE OR ROUND-OFF PREVENTS
0023              ATTAINING REQUIRED ACCURACY
0024
0025      C R. H. FRENCH, 14 AUGUST 1984
0026
0027      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0028      EXTERNAL F
0029      DIMENSION WORK(N),STACK(N),HEAP(N)
0030      KODE=0
0031      Y=0.00
0032      WORK(1)=XU
0033      CALL QR(XL,XU,F,T)
0034      HEAP(1)=T
0035      A=XL
0036      NPTS=1
0037      EPS=TOL
0038      STACK(1)=EPS
0039      B=WORK(NPTS)
0040      XM=(A+B)*0.5D0
0041      CALL QR(XL,XM,F,P1)
0042      CALL QR(XM,B,F,P2)
0043      CALL DABS(T-P1-P2).LE.EPS) GOTO 20
0044
0045      C SPLIT IT
0046      NPTS=NPTS+1
0047      IF(NPTS.GT.N) THEN
0048          KODE=-1
0049          RETURN
0050      END IF
0051      WORK(NPTS)=XM
0052      HEAP(NPTS)=P2
0053      EPS=EPS/2.0D0

```

PDP-11 FORTRAN-77 V4.0-1 13:13:34
 CLIP2HOP.FTM;14 /F77/MR 16-Jul-86

```

0028 IF(EPS.EQ.0.00) THEN
0029   KODE=2
0030   RETURN
0031 END IF
0032 STACK(NPTS)=EPS
0033 GOTO 10
0034 C FINISHED A PIECE
0035 Y=Y+P1+P2
0036 EPS=STACK(NPTS)
0037 T=MEXP(NPTS)
0038 NPTS=NPTS-1
0039 A=B
0040 IF(NPTS.EQ.0) RETURN
0041 GOTO 10
0042 END

```

Page 25

PDP-11 FORTRAN-77 V4.0-1 13:13:37
 CLIP2HOP.FTM;14 /F77/MR 16-Jul-86

Page 26

```

0001 SUBROUTINE D616(A,B,F,ANSWER)
C 16-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL
C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4
C R. H. FRENCH, 28 FEBRUARY 1986
C
0002 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON /WTS/ X(8),W(8)
ANSWER=0. DO
0003 BWA02=(B-A)/2.00
0004 BPA02=(B+A)/2.00
0005 DO 10 I=1,8
0006 C=X(I)*BWA02
0007 C4=(I)*BWA02
0008 Y1=BPA02+C
0009 Y2=BPA02-C
0010 ANSWER=ANSWER+W(I)*(F(Y1)+F(Y2))
0011 CONTINUE
0012 10 ANSWER=ANSWER*BWA02
0013 RETURN
0014
0015 END

```

H-13

PDP-11 FORTRAN-77 V4.0-1 13:13:38
 CLIP2HOP.FTM;14 /F77/MR 16-Jul-86

Page 27

```

0001 SUBROUTINE D6XVI(A,B,F,ANSWER)
C 16-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL
C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4
C R. H. FRENCH, 28 FEBRUARY 1986
C
0002 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON /WTS/ X(8),W(8)
ANSWER=0. DO
0003 BWA02=(B-A)/2.00
0004 BPA02=(B+A)/2.00
0005 DO 10 I=1,8
0006 C=X(I)*BWA02
0007 C4=(I)*BWA02
0008 Y1=BPA02+C
0009 Y2=BPA02-C
0010 ANSWER=ANSWER+W(I)*(F(Y1)+F(Y2))
0011 CONTINUE
0012 10 ANSWER=ANSWER*BWA02
0013 RETURN
0014
0015 END

```

PDP-11 FORTRAN-77 V4.0-1 13:13:40 16-Jul-86
CLIP2HOP.FTN;14 /F77/MR

Page 28

PDP-11 FORTRAN-77 V4.0-1 13:13:43 16-Jul-86
CLIP2HOP.FTN;14 /F77/MR

Page 30

0001 C SUBROUTINE TEST(1D)
C TEST RETURN CODE FROM BESSEL FUNCTION
C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
IF(KODE.EQ.0) RETURN
WRITE(5,1) KODE, ID
1 FORMAT(' BESSEL FUNCTION CODE = ',I2,' FROM CALL NUMBER ',I5)
STOP 'FATAL ERROR.'
END

PDP-11 FORTRAN-77 V4.0-1 13:13:42 16-Jul-86
CLIP2HOP.FTN;14 /F77/MR

Page 29

0001 C SUBROUTINE TEST2(KODE,1D)
C TEST RETURN CODE FROM ADQUAD/ADQUA2
C IF(KODE.EQ.0) RETURN
WRITE(5,1) KODE, ID
1 FORMAT(' ADAPTIVE INTEGRATOR CODE = ',I2,' FROM CALL NUMBER ',I5)
STOP 'FATAL ERROR.'
END

4

0001 C DOUBLE PRECISION FUNCTION P21(Y)
C SIGNAL CHANNEL P.D.F. WITH CHANGE OF VARIABLE Y-AX
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION WORK(30), STACK(30), HEAP(30)
LOGICAL*1 REG1, REG2
EXTERNAL DXYT1, F20, F21, F22
COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
COMMON /DENPAR/ BIGK, AAB, LJAM,
TAU, TAU2, TAUK, TAUK2
COMMON /PARDEN/ RHOM, RHOT1, RHOT2, RHOMA, RHOT4,
RHOMB, RHOTB, RHOMH, RHOTH
COMMON /QUES/ QQ, Q1
COMMON /XCON/ XCON
COMMON /OUTER/ XXX, XXK
XXI=Y
IF(LJAM.GE.1) THEN
XXX=V/BIGK
YK=XXK
YTK=(Y-TAU)/BIGK
YTK2=(Y-TAU2)/BIGK
END IF
REG1=Y GE.0 DO .AND. Y LT. TAU
REG2=Y GE. TAU .AND. Y LT. TAU2
C TWO HOPS PER SYMBOL
C
0002 2100 IF(REG1) THEN
BARG1=DSQR(TRHOMB*Y)
101=1
0023 CALL DXBT(2100)
0024 P21=0.5D0*DSQRT(Y/RHOMH)*DEXP(BARG1-Y-RHOM2)*B1
0025 ELSE IF(REG2) THEN
BARG1=DSQR(TRHOM4*(Y-TAU))
101=0
0026 CALL DXBT(2101)
0027 PART=2.0D0*DEXP(BARG1-Y-TAU-RHOM)*B1
0028 XCON=Y+RHOM2
0029 CALL ADQUA2(Y-TAU,TAU,ANSWER,DERIV,F20,1,D-9,WORK,STACK,
HEAP,30,KOD)
0030 CALL TEST2(KOD,2100)
0031 P21=PART+ANSWER
0032 0036 ELSE
0033 P21=0.00
0037 END IF
0038 GO TO 9000
0039

```

C          ONE HOP JAMMED
C
0040      2200  IF( REG1 ) THEN
0041          XCON=YK+RHOM+RHO1
0042          CALL ADQUA2(0.00,Y,ANSWER,DGIV1,F21,1,D-9,WORK,STACK,
0043          HEAP,30,K00)
0044          $   CALL TEST2(K00,2200)
0045          PZ1=ANSWER/B1GK
0046          ELSE IF( REG2 ) THEN
0047              BARG1=DSQRT( RHO1*YK )
0048              101=0
0049              CALL DXBT(1201)
0050              PART=0.0*DEXP( BARG1-YK+TAU2-RHO1)*B1/B1GK
0051              BARG1=DSQRT( RHOM*( Y-TAU1 ) )
0052              101=0
0053              CALL DXBT(1202)
0054              PART=PART+0.1*DEXP( BARG1-Y+TAU1-RHOM )*B1
0055              XCON=YK+RHOM+RHO1
0056              CALL ADQUA2( Y-TAU1,TAU1,ANSWER,DGIV1,F21,1,D-9,WORK,STACK,
0057              HEAP,30,K00)
0058              PZ1=PART+ANSWER/B1GK
0059              ELSE
0060                  PZ1=0.00
0061              END IF
0062              GOTO 9000
C          TWO HOPS JAMMED
C
0062      2300  IF( REG1 ) THEN
0063          BARG1=DSQRT( RHO1*YK )
0064          101=1
0065          CALL DXBT(12301)
0066          PZ1=0.50*DSQRT( YK/RHOM )*DEXP( BARG1-YK-RHO1 )
0067          *B1/B1GK
0068          ELSE IF( REG2 ) THEN
0069              BARG1=DSQRT( RHO1*YK )
0070              101=0
0071              CALL DXBT(12301)
0072              PART=2.0*PZ1*DEXP( BARG1-YK-RHO1 )*B1/B1GK
0073              XCON=YK+RHO1
0074              CALL ADQUA2( Y-TAU1,TAU1,ANSWER,DGIV1,F22,1,D-9,WORK,STACK,
0075              HEAP,30,K00)
0076              CALL TEST2(K00,2300)
0077              PZ1=PART+ANSWER/( B1*REG1 )
0078              PZ1=0.00
0079              END IF
0080              CONTINUE
0081              RETURN
C          ONE HOP JAMMED
C
0001      C  DOUBLE PRECISION FUNCTION GL(Y)
C  C  NON-SIGNAL CHANNEL CUMULATIVE DISTRIBUTION FUNCTION
C  C  WITH CHANGE OF VARIABLE Y-AX
0002      C  IMPLICIT DOUBLE PRECISION ( A-H,O-Z )
0003      LOGICAL*1 REG1, REG2, REG3
0004      COMMON /DENPAR/ B1GK, AAB, BAB, LJMM,
0005      $   REG1=Y, GE, 0.00 , AND, YLT, TAU
0006      REG2=Y, GE, TAU , AND, YLT, TAU2
0007      IF( LJMM .GT. 0) THEN
0008          YK=Y/B1GK
0009          YTK=(Y-TAU)/B1GK
0010          YT12=(Y-TAU2)/B1GK
0011      END IF
C  C  TWO HOPS PER SYMBOL
0012      C
0013      C  IF( REG1 ) THEN
0014          GL=1.00-(1.00*Y)*DEXP( -Y )
0015          ELSE IF( REG2 ) THEN
0016              GL=1.00-(1.00+TAU2-Y)*DEXP( -Y )
0017          ELSE IF( Y .GE. TAU2 ) THEN
0018              GL=1.00
0019          ELSE
0020              GL=0.00
0021          END IF
0022          GOTO 9000
C  C  ONE HOP JAMMED
C
0023      2200  IF( REG1 ) THEN
0024      GL=1.00-AAB*DEXP( -YK )+BAB*DEXP( -Y )
0025      ELSE IF( REG2 ) THEN
0026      GL=1.00-AAB*DEXP( TAU-TAU2-Y )+BAB*DEXP( TAU-K-TAU-YK )
0027      ELSE IF( Y .GE. TAU2 ) THEN
0028      GL=1.00
0029      ELSE
0030      GL=0.00
0031      END IF
0032      GOTO 9000

```

PDP-11 FORTRAN-77 V4.0-1 /F77/MR
CLIP2HOP.FTN;14

Page 33

PDP-11 FORTRAN-77 V4.0-1 /F77/MR
CLIP2HOP.FTN;14

Page 35

C

C TWO HOPS JAMMED

```
0033 2300 IF(REQ1) THEN
0034   GL=1.0D-(1.0D+YK)*DEXP(-YK)
0035   ELSE IF(REQ2) THEN
0036     GL=1.0D-(1.0D+TAUK2-YK)*DEXP(-YK)
0037   ELSE IF(Y_GE.TAU2) THEN
0038     GL=1.0D
0039   ELSE
0040     GL=0.0D
0041   END IF
0042   GOTO 9000
0043   CONTINUE
0044   RETURN
0045   END
```

PDP-11 FORTRAN-77 V4.0-1 /F77/MR
CLIP2HOP.FTN;14

Page 34

```
0001      DOUBLE PRECISION FUNCTION F21(U)

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=2, L_JAH=1
C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
COMMON /DEMPAR/ BARG1, BARG2, B1, B2, KODE, 101, 102
COMMON /DENPAR/ BICK, AAB, BAB, LJAM,
S COMMON /PARDEN/ RHOM, RHOT, RHOT2, RHOT4,
S COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
COMMON /QUES/ QO, Q1
COMMON /XCON/ XCON
COMMON /OUTER/ XXX, XXXK
BARG1=DSORT(RHOM4*U)
BARG2=DSORT(RHOT4*(XXXK-U/BIGX))
101=0
102=0
CALL BPROD(2110)
F21=EXP(BARG1+BARG2-XCON-U+U/BIGX)*B1*B2
RETURN
END
```

PDP-11 FORTRAN-77 V4.0-1 /F77/MR
CLIP2HOP.FTN;14

Page 35

```
0001      DOUBLE PRECISION FUNCTION F21(U)

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=2, L_JAH=1
C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
COMMON /DEMPAR/ BARG1, BARG2, B1, B2, KODE, 101, 102
COMMON /DENPAR/ BICK, AAB, BAB, LJAM,
S COMMON /PARDEN/ RHOM, RHOT, RHOT2, RHOT4,
S COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
COMMON /QUES/ QO, Q1
COMMON /XCON/ XCON
COMMON /OUTER/ XXX, XXXK
BARG1=DSORT(RHOM4*U)
BARG2=DSORT(RHOT4*(XXX-U))
101=0
102=0
CALL BPROD(2110)
F21=EXP(BARG1+BARG2-XCON-U+U/BIGX)*B1*B2
RETURN
END
```

PDP-11 FORTRAN-77 V4.0-1 /F77/MR
CLIP2HOP.FTN;14

Page 33

Page 35

```
0001      DOUBLE PRECISION FUNCTION F21(U)

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=2, L_JAH=1
C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
COMMON /DEMPAR/ BARG1, BARG2, B1, B2, KODE, 101, 102
COMMON /DENPAR/ BICK, AAB, BAB, LJAM,
S COMMON /PARDEN/ RHOM, RHOT, RHOT2, RHOT4,
S COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
COMMON /QUES/ QO, Q1
COMMON /XCON/ XCON
COMMON /OUTER/ XXX, XXXK
BARG1=DSORT(RHOM4*U)
BARG2=DSORT(RHOT4*(XXX-U))
101=0
102=0
CALL BPROD(2110)
F21=EXP(BARG1+BARG2-XCON-U+U/BIGX)*B1*B2
RETURN
END
```

Page 36

PDP-11 FORTRAN-77 V4.0-1
CLJP2HOP.FTN;14 /F77/NR

Page 37

```
0001      C DOUBLE PRECISION FUNCTION F22(I)
          C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, I=2, LJM=2
          C
          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          COMMON /DEMPAR/ B1GK, AAB, BAB, LJMM,
          S   TAU, TAU2, FAIR, TAUK2
          S   COMMON /PARDEN/ RHON, RHOT, RHOM2, RHOT2, RHOM4, RHOT4,
          S   RHOMB, RHOTB, RHOMH, RHOHT, RHOM2, RHOT2, RHOM4, RHOT4,
          S   COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
          S   COMMON /QUES/ Q0, Q1
          S   COMMON /ACCM/ XCOM
          S   COMMON /OUTER/ XXX, XUXX
          S   BARG1=DSQRT(RHOT4*U/B1GK)
          S   BARG2=DSQRT(RHOT4*(XXX-U/B1GK))
          101=0
          102=0
          CALL BPROD(2310)
          F22=DEXP(BARG1+BARG2-XCOM)*B1*82
          RETURN
          END
```

T-17 PDP-11 FORTRAN-77 V4.0-1 13:14:01 16-Jul-86

```
0001      C SUBROUTINE DABT(I0)
          C CALL DBEST1 AND TEST RETURN CODE
          C
          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
          CALL DBEST1(BARG1,101,B1,KODE)
          CALL TEST(I0)
          RETURN
          END
```

Page 38

PDP-11 FORTRAN-77 V4.0-1
CLJP2HOP.FTN;14 /F77/NR

Page 39

```
0001      C SUBROUTINE TIES(JSUB,NN,PTIE)
          C COMPUTE PROBABILITY OF CORRECT DECISION GIVEN THAT
          C SEVERAL SATURATED CHANNELS ARE TIED
          C
          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          DIMENSION JSUB(NN), LL0W(7), LINC(7), LUP(7), MU(7),
          S   P2LN(2:8)
          LOGICAL I60
          COMMON /DEMPAR/ B1GK, AAB, BAB, LJMM,
          S   TAU, TAU2, TAUK, TAUK2
          S   COMMON /PARDEN/ RHON, PHOT, RHOM2, RHOT2, RHOM4, RHOT4,
          S   RHOMB, RHOTB, RHOMH, RHOHT, RHOM2, RHOT2, RHOM4, RHOT4,
          S   COMMON /QUES/ Q0, Q1
          S   COMMON /NON-SIGNAL CHANNELS/
          MN0+NN-1
          PTIE=0.00
          CUEI=DEXP(-TAUK)
          CUEI=DEXP(-TAUK)
          P1I=DXI(00,2,JSUB(1))*DXI(Q1,JSUB(1))
          DO 10 I=2,NN
          P2LN(1)=DXI(CUE0,2,JSUB(1))*DXI(CUE1,JSUB(1))
          10 CONTINUE
          0009
          0010
          0011
          0012
          0013
          0014
          0015
          0016
          0017
          0018
          0019
          0020
          0021
          0022
          0023
          0024
          0025
          0026
          0027
          0028
          0029
          0030
          0031
          0032
          0033
          0034
          0035
          0036
          0037
          0038
          0039
          0040
          0041
          C SET UP VECTOR LOOP PARAMETERS
          DO 20 I=1,NN-1
          LL0W(I)=0
          LINC(I)=1
          LUP(I)=1
          20 CONTINUE
          PTIE=0.00
          C START LOOP ON THE TIE EVENTS
          CALL VLINT(MU,LL0W,MN0-1)
          30   MU0M=0
          DO 40 I=1,NN-1
          MU0M=MU0M+MU(I)
          40 CONTINUE
          FRAE=1.00/(1.00+MU0M)
          P00=1.00
          DO 50 MN=2,NN
          IF(MU(I-1).EQ.1) THEN
          P00=P00*P2LN(M)
          ELSE
          P00=P00*(1.00-P2LN(M))
          END IF
          CONTINUE
          PTIE=FRAE*P00+PTIE
          CALL VLITER(MU,LL0W,LINC,MN-1,GO)
          IF(GO) GOTO 30
          PTIE=PTIE*P1L
          RETURN
          END
```

PDP-11 FORTRAN-77 V4.0-1 13:14:02 16-Jul-86

POP-11 FORTRAN-77 V4.0-1 13:14:07 16-Jul-86
 CLIP2HOP.FTN:14 /F77/WR

Page 40

POP-11 FORTRAN-77 V4.0-1 13:14:08 16-Jul-86
 CLIP2HOP.FTN:14 /F77/WR

Page 41

```

0001      C DOUBLE PRECISION FUNCTION PUNJAM(ETA)
          C FUNCTION FOR UNJAMMED P(E) FOR OPT. THRESHOLD SEARCH
          C
          C NOTE: WHEN JAMMING EVENT IS (0,0,...,0), THE VARIABLES
          C     BIGK, AAB, BAB, TAUK, AND TAUK2 ARE NOT
          C     USED IN THE COMPUTATIONS, AND HENCE DO NOT NEED
          C     TO BE SET UP BEFORE CALLING PSEL FROM THIS FUNCTION
          C
          C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          C DIMENSION NOJAM(4)
          C COMMON /INPUTS/ DEBNDL(3),MSLOTS,GAMLST(10),K,MM
          C COMMON /DENPAR/ BIGK,AAB,BAB,LJAM,
          C                 TAU,TAU2,TAUK,TAUK2
          C
          C DATA NOJAM/0,0,0,0/
          LJAM=0
          TAU=ETA
          TAU2=TAU+ETA
          CALL PSEL(NOJAM,MM,P)
          PUNJAM=P
          RETURN
          END
0013

```

```

0002      C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          C SEARCH FOR OPTIMUM THRESHOLD IN ABSENCE OF JAMMING
          C
          C EXTERNAL PUNJAM
          C IMPLICIT DOUBLE PRECISION(A-H,O-Z);
          C
          C SUBROUTINE SETTAU(MM,PE00)
          C
          C
          C 0001      C SEARCH FOR OPTIMUM THRESHOLD IN ABSENCE OF JAMMING
          C
          C 0002      C IMPLICIT DOUBLE PRECISION(A-H,O-Z);
          C
          C 0003      C EXTERNAL PUNJAM
          C
          C 0004      C COMMON /DENPAR/ BIGK,AAB,BAB,LJAM,
          C                 TAU,TAU2,TAUK,TAUK2
          C
          C 0005      C COMMON /PARDEN/ RHON,RHOT,RHOT2,RHOMH,RHOTH
          C
          C 0006      C COMMON /QUES/ QQ,Q1
          LJAM=0
          C
          C 0007      C GUESS BASED ON QUADRATIC CURVE FIT
          C
          C IF (MM.EQ.2) THEN
          C     GUESS=0.925D0*4.8-475D0*2+32.45D0
          C ELSE IF (MM.EQ.4) THEN
          C     GUESS=1.050D*4-9.350D*2+34.50D
          C ELSE IF (MM.EQ.8) THEN
          C     GUESS=1.100D*4-9.90D*2+36.30D
          C ELSE
          C     GUESS=15.00
          C END IF
          C CALL MINSER(PUNJAM,PENIN,TAUOPT,1.00,GUESS,0.00)
          C
          C 0018      C TAU=TAUOPT
          C 0019      C TAU2=TAU+TAU
          C 0020      C PE0=PERIN
          C 0021      C RETURN
          C 0022

```

PDP-11 FORTRAN-77 V4.0-1 13:14:10 16-Jul-86 Page 42
 CLIP2HOP.FTN.14 /F77/HB

0001 C SUBROUTINE MINSER(F,FMIN,XMIN,STEP,GUESS,BLIM,ULIM,TOL)
 C SEARCH FOR MINIMUM OF F(X) OVER THE INTERVAL BLIM <= X <= ULIM
 C TROUBLE MAY OCCUR IF F(X) HAS MULTIPLE LOCAL MINIMA WITHIN THE
 C SEARCH INTERVAL OR IF THE FUNCTION IS VERY STEEP AND STEP IS
 C TOO BIG.
 C

C F = NAME OF FUNCTION TO BE MINIMIZED
 C FMIN = MINIMUM VALUE OF F(X) OVER INTERVAL
 C XMIN = VALUE OF X FOR WHICH FMIN OCCURS
 C STEP = INITIAL STEP SIZE FOR SEARCH
 C GUESS = INITIAL GUESS AT XMIN, BLIM <= GUESS <= ULIM
 C BLIM = LOWER LIMIT OF SEARCH INTERVAL
 C ULIM = UPPER LIMIT OF SEARCH INTERVAL
 C TOL = TOLERANCE ON XMIN; SEARCH STOPS WHEN DX < TOL
 C
 C NOTE: F MUST BE A DOUBLE PRECISION FUNCTION OF ONE
 C DOUBLE PRECISION ARGUMENT. ANY PARAMETERS CAN BE PASSED
 C FROM THE CALLER VIA A COMMON BLOCK.

C PROGRAMMER: ROBERT H. FRENCH DATE: 17 MARCH 1986

C
 C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
 C
 1-0003 X=GUESS
 0004 SUMAX=DABS(X-BLIM)
 0005 SUMAX=DABS(ULIM-X)
 0006 DX=DMIN1(STEP,SUMAX,SUMAX)
 0007 TEST=TOL
 0008 10 F0=F(X)
 0009 F1=F(X+DX)

C ARE WE GOING IN THE RIGHT DIRECTION?
 0010 10 IF(F1.LE.F0) GOTO 100
 C ... NO, SWITCH DIRECTION
 0011 DX=-DX
 0012 F1=F(X+DX)
 0013 IF(F1.LE.F0) GOTO 100

C ELSE WE MUST BE CLOSE TO A MIN. AT X=GUESS, SO CUT
 C STEP SIZE AND TRY AGAIN
 0014 DX=DX/10.0D0
 0015 IF(DABS(DX).GE.TEST) GOTO 10

C CLOSE ENOUGH AT GUESS
 0016 12 XMIN=X

FMIN=F0

RETURN

C NOW GOING RIGHT DIRECTION. KEEP GOING UNTIL PAST MINIMUM
 C BY ONE STEP.
 C

0019 100 X2=X+DX
 C HAVE WE REACHED END POINT?
 0020 IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110

PDP-11 FORTRAN-77 V4.0-1 13:14:10 16-Jul-86 Page 43
 CLIP2HOP.FTN.14 /F77/HB

0021 C ALL OK
 105 F2=F(X2)
 C PAST MIN?
 0022 IF(F2.GE.F1) GOTO 200
 C ...NO, STEP AGAIN
 F0=F1
 F1=F2
 X=X+DX
 GOTO 100
 C MIN MAY BE AT AN ENDPOINT. CUT STEP SIZE AND TRY AGAIN
 C IF INCREMENT NOT TOO SMALL.
 110 IF(DABS(DX).LE.TEST) GOTO 120
 X=X+DX
 GOTO 100
 C MIN MUST BE AT THE ENDPOINT (OR WITHIN MINIMUM DX THEREOF)
 115 DX=DX/10.0D0
 F1=F(X+DX)
 GOTO 100
 C MIN AT X-ULIM
 XMIN=ULIM
 FMIN=F(XMIN)
 RETURN
 C MIN AT BLIM
 122 XMIN=BLIM
 GOTO 121
 C HAVE PASSED MIN. IS IT LOCATED CLOSELY ENOUGH YET?
 200 IF(DABS(DX).LE.TEST) GOTO 300
 C ... NO, CUT STEP SIZE AND TRY AGAIN
 0028 GOTO 115
 C DONE!
 C SINCE F0 >= F1 & ABS(DX)<MIN. DX, CALL F1 THE MIN.
 300 FMIN=F1
 XMIN=X+DX
 RETURN
 END

0029 F0=F1
 0030 115 DX=DX/10.0D0
 0031 F1=F(X+DX)
 0032 GOTO 100
 C MIN AT X-ULIM
 0033 120 IF(X2.LE.BLIM) GOTO 122
 C MIN AT X-ULIM
 0034 XMIN=ULIM
 0035 121 FMIN=F(XMIN)
 0036 RETURN
 C MIN AT BLIM
 0037 122 XMIN=BLIM
 0038 GOTO 121
 C HAVE PASSED MIN. IS IT LOCATED CLOSELY ENOUGH YET?
 0039 200 IF(DABS(DX).LE.TEST) GOTO 300
 C ... NO, CUT STEP SIZE AND TRY AGAIN
 0040 GOTO 115
 C DONE!
 C SINCE F0 >= F1 & F2>= F1 AND ABS(DX)<MIN. DX, CALL F1 THE MIN.
 0041 300 FMIN=F1
 0042 0043 0044
 END

0045 FMIN=F0
 RETURN

H-20

J. S. LEE ASSOCIATES, INC.

APPENDIX I COMPUTER PROGRAM FOR CLIPPER RECEIVER FOR M=4 AND L=2

The following pages contain the source code listing for the FORTRAN-77 program used for numerical computations of the performance of the clipper receiver for the case of M=4 and L=2 with a numerical search for the worst-case jamming fraction. By increasing the array A to 256 elements and the arrays C and D to 6561 elements each, and changing the array size parameters to calls to PUTIN and LOOKUP, the program may also be used for the case M=8, L=2.



```

0001      PROGRAM CLPAN
C THIS PROGRAM COMPUTES THE ERROR PROBABILITY FOR RANDOM 4-ARY
C FSK/FH WITH 2 HOPS/BIT IN THE PRESENCE OF PARTIAL-BAND
C NOISE JAMMING BY NUMERICAL INTEGRATION FOR THE CLIPPER RECEIVER
C ANALYSIS: L. E. MILLER, R. H. FRENCH
C PROGRAM: R. H. FRENCH
C
C V 2.1.0 - COMPUTATIONS ONLY
C
C
0002      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003      PARAMETER (LJ=51)
0004      CHARACTER*1 YES, NO, REPLY, BLANK
0005      CHARACTER*13 FNAME, GNAME
0006      LOGICAL DOTAU, TEST
0007      LOGICAL *1 GOOD
0008      REAL*4 PRLOG(LJ), DBSJRL(LJ), QOPT(LJ)
0009      VIRTUAL A1(100), TAU(100), C(625), PCSUB(625)
0010      VIRTUAL D(625), IDSUB(625), PRSR(625), JPSSUB(625)
0011      C COMMON /INPUTS/ PASSES PARAMETERS OF THE RUN
0012      C COMMON /INPUTS/ PASSES PARAMETERS OF THE RUN
0013      C COMMON /SIZE/ PASSES NUMBERS OF PARAMETERS
0014      COMMON /DENPAR/ BIGX, AAB, BAB, LJMM,
0015          TAU, TAU2, TAUK, TAUK2
0016      COMMON /PARDEN/ RHOM, RIOT
0017      DATA YES, NO, BLANK /'Y', 'N', ' '/
0018      CALL ERSET2S(.TRUE.,.FALSE.,.TRUE.,.FALSE.,.15)
0019      CALL GET(MJ,START,DBINC)
0020      SLOTS=MSLOTS
0021      WERBIT=0.500*RHOM/(MM-1.00)
0022      DO 800 10=1,NO
0023      DOTAU=.TRUE.
0024      EBNO=10.00**((DEBNOL(10)/10.00)
0025      RHOM=K*EBNO/2.00
0026      LOGOUT=DEBNOL(10)
C OPEN DATA FILE
C
0026      IGINUT=Gamma*1000.00+0.500
0027      WRITE(FNAME,730) MM,1000T
0028      730      FORMAT('COJ',IJ,'2',12,2,'.DAT')
0029      WRITE(6,776) MM,DEBNOL(10)
0030      776      FORMAT('CLIPPER? RECEIVER, OPTIMUM GAMMA RESULTS','/
0031          ,MM-'1.5X,L=2,'5X,EB/N0=','F8.4//','EB/NJ (dB)',/
0032          ,5X,'P(e)'15X,'Opt')
0033      WRITE(5,733) FNAME
0034      FORMAT(' WORKING ON ',A13)
0035      OPEN(UNIT=4,FILE=FNAME,STATUS='OLD',FORM='UNFORMATTED',
0036          ERR=750)
0036      $ DEBNOL=START+(IJ-1)*DBINC
C HAVE AN EXISTING FILE, READ TO SEE HOW FAR WE GOT BEFORE
C WE HAVE READ A VALUE OF TAU, SO WE WON'T NEED TO RECOMPUTE
C IT UNTIL EITHER EB/NO, MM, OR LL CHANGES
C
0034      1300      READ(4,END=742) DBSAR(JJ), PRLOG(JJ), QOPT(JJ)
0035      DOTAU=.FALSE.
0036      JJ=0
0037      740      READ(4,END=742) DBSAR(JJ), PRLOG(JJ), QOPT(JJ)
0038      0038      GOTO 740
0039      0040      CLOSE(UNIT=4)
0041      0041      GOTO 755
C NO EXISTING FILE, THIS IS THE FIRST TIME:
C CREATE FILE HEADER RECORD
C
0042      750      JJ=1
C ... WE CAN'T READ TAU FROM A FILE, SO IF THE FIRST TIME
C THROUGH THE LOOP ON EB/NO WE MUST COMPUTE IT. BUT IF
C EB/NO HASN'T CHANGED, WE DON'T NEED TO RECOMPUTE IT SINCE
C THE THRESHOLD IS NOT A FUNCTION OF GAMMA NOR OF EB/NO.
C
0043      IF(DOTAU) THEN
0044          WRITE(5,757)
0045          757      FORMAT(' SETTING THRESHOLD')
0046          CALL SETTAU(MM,PE00)
0047          DOTAU=.FALSE.
0048          WRITE(5,1991) MM, TAU
0049          WRITE(6,1991) MM, TAU
0050          0050      FORMAT(' M=' ,IJ, ' L=' ,L-2, ' OPT THRES = ',1PD15.8)
0051          END IF
0052      OPEN(UNIT=6,FILE=FNAME,STATUS='NEW',FORM='UNFORMATTED')
0053      WRITE(4,MM,DEBNOL(10),MSLOTS,TAU,TAU2,PE00)
0054      CLOSE(UNIT=4)
0055      0055      DO 600 IJ=JJ,MM
0056      IF(IJ.GE.3) THEN
0057          IQQ=QOPT(IJ-1)-QOPT(IJ-2)*0.500
0058          IF(IJQ,0.0,0) 100=1
0059          0059      Q=QOPT(IJ-1)
0060          0060      IQ=0
0061          0061      ELSE
0062          0062      IQ=1
0063          0063      Q=1.00
0064          0064      IQ=1
0065          0065      END IF
0066          0066      DO=100
C GIVE PROGRESS MESSAGE TO TI:
0067          0067      WRITE(5,60) IJ
0068          0068      FORMAT(' I.=',IJ)
0069          0069      DEBNOL=START+(IJ-1)*DBINC

```

```

0070      DBSR(1J)=DEBNJ
          DO 602 1,JO=1,2400
0071      602      PESAV(1,JO)=0.00
          R=10.00**DEBNJ/10.00
0072      C PRIME THE ALGORITHM WITH DUMMY OLD VALUES OF P(E)
          P1=0.00
          P2=0.00
0073      709      GAMMA=0/MSLOTS
          WRITE(GNAME,735) MM,1Q
0074      735      FORMAT('EQ',1L,'2','14.4','DAT')
          OPEN(UNIT=3,FILE=GNAME,STATUS='OLD',FORM='UNFORMATTED',
     *      READONLY,ERR=770)
          $      WRITE(5,3939)
0075      3939     FORMAT(' READING EVENT FILE')
          READ(3,D10SUB,MUSED,6000)
          CLOSE(UNIT=3)
          GOTO 777
0076      C IF FILE FOR EVENT PROBABILITIES DOES NOT EXIST, CALCULATE THEM
          C AND CREATE A FILE.
0077      770      CONTINUE
          WRITE(5,3938)
0078      3938     FORMAT('1. CREATING EVENT FILE')
          CALL GENPIE(MM,1Q,MSLOTS,GOOD,A,JASUB,C,1CSUB,
     *      0,1DSUB,MUSED)
          $      OPEN(UNIT=3,FILE=GNAME,STATUS='NEW',FORM='UNFORMATTED')
          WRITE(3,D10SUB,MUSED,GOOD)
          CLOSE(UNIT=3)
0079      777      IF(.NOT.GOOD) GOTO 700
          RHOT=S*GAMMA*R*EBNO/(GAMMA*R*EBNO)
          RHOT=R*RHOTS/2.00
0080      C EVALUATE THE PROBABILITY
          CALL PSUB(MM,PESYM,D,1DSUB,MUSED,PRERR,FPSUB,P00,
     *      PESAV,1Q)
          $      P3=PESTM
          IF(P3.GT.P2 .AND. 1Q.LT.MSLOTS) THEN
0081      C KEEP ON GOING, WE ARE NOT PAST THE MAXIMUM
          P1=P2
          P2=P3
          1Q=MIN(1Q+100,MSLOTS)
          Q=MIN(1Q+DQ,MSLOTS)
          GOTO 709
0082      ELSE
          PHMAX=DMAX1(P1,P2,P3)
          EPS=0.001D0*PHMAX
          TEST=(DABS(P1-P2).LE.EPS .AND. DABS(P1-P3).LE.EPS .AND.
0083      $      DABS(P2-P3).LE.EPS)
          IF( TEST .OR. 1D0.EQ.1
0084      $      .OR. (1..NOT.TEST) .AND. 1Q.EQ.MSLOTS) THEN
          C WE ARE DONE WHEN ALL 3 ARE CLOSE TOGETHER OR WHEN DQ=1
          C OR WHEN WE REACHED FULL-BAND JAMMING AND P(E) IS STILL
          C INCREASING
          POPT=PHMAX

```

```

0109      C
0110      0111      C PREVENT ROUND-OFF FROM MAKING QOPT VS. EB/MJ NON-MONOTONIC
0112      IF(1J.GT.1) THEN
          IF(QOPT(1J).LT.QOPT(1J-1)) QOPT(1J)=QOPT(1J-1)
          END IF
          ELSE
          THE OPTIMUM MUST BE THE MIDDLE POINT OF THE 3
          QOPT(1J)=Q-DQ
          IF(QOPT(1J).EQ.0.00) QOPT(1J)=1.00
          END IF
          IF(1J.GT.1) THEN
          IF(QOPT(1J).LT.QOPT(1J-1)) QOPT(1J)=QOPT(1J-1)
          END IF
          ELSE
          THE OPTIMUM IS FULL-BAND JAMMING
          QOPT(1J)=MSLOTS
          END IF
          GOTO 665
0113      C
0114      0115      C NOT LOCATED SUFFICIENTLY ACCURATELY. CUT DO AND TRY AGAIN
0116      0117      C
0117      0118      C
0118      0119      C
0119      0120      C
0121      0122      C
0122      7=100/2
0123      D=1Q
0124      P2=P1
0125      P1=0.00
0126      0=0+DQ
0127      1Q=1Q+DQ
0128      GOTO 709
0129      END IF
0130      END IF
0131      665      P=WORKB1*POPT
          WRITE(6,666) DBSR(1J),PE,QOPT(1J)
          FORMAT(6X,F7.3*X,1P012.5,5X,1P012.5)
0132      666      PRLOG(1J)=DBLOG(PE)
          OPEN(UNIT=4,FILE=FNAME,STATUS='OLD',ACCESS='APPEND',
     *      FORM='UNFORMATTED')
          $      WRITE(4,653R(1J),PRLOG(1J),QOPT(1J))
          CLOSE(UNIT=4)
0133      666      OPEN(UNIT=4,FILE=FNAME,STATUS='NEW',FORM='UNFORMATTED')
          WRITE(4,653R(1J),PE,QOPT(1J))
          CLOSE(UNIT=4)
          WRITE(6,776) MM,DEBML(10),MSLOTS,DBSR,PRLOG,QOPT
0134      666      WRITE(6,776) MM,DEBML(10)
          DO 689 1J=1,MJ
          WRITE(6,666) DBSR(1J),10.*PRLOG(1J),PE,QOPT(1J)
0135      689      CONTINUE
          WRITE(6,668) TAU
          FORMAT(6X,'OPTIMUM THRESHOLD FOR ABOVE IS ETA/SIGMA**2 ',F7.3)
0136      688      $      CONTINUE
0137      600      CONTINUE
          OPEN(UNIT=4,FILE=FNAME,STATUS='NEW',FORM='UNFORMATTED')
          WRITE(4,653R(1J),PE,QOPT(1J))
          CLOSE(UNIT=4)
          WRITE(6,776) MM,DEBML(10)
          DO 689 1J=1,MJ
          WRITE(6,666) DBSR(1J),10.*PRLOG(1J),PE,QOPT(1J)
0138      600      CONTINUE
          CONTINUE
0139      600      CONTINUE
          WRITE(4,653R(1J),PE,QOPT(1J))
          CLOSE(UNIT=4)
          WRITE(6,776) MM,DEBML(10)
          DO 689 1J=1,MJ
          WRITE(6,666) DBSR(1J),10.*PRLOG(1J),PE,QOPT(1J)
0140      600      CONTINUE
          CONTINUE
0141      640      CONTINUE
          WRITE(6,776) MM,DEBML(10)
          DO 689 1J=1,MJ
          WRITE(6,666) DBSR(1J),10.*PRLOG(1J),PE,QOPT(1J)
0142      640      CONTINUE
          CONTINUE
0143      640      CONTINUE
          STOP 0
0144      640      CONTINUE
          STOP 0
0145      689      CONTINUE
          WRITE(6,668) TAU
          FORMAT(6X,'OPTIMUM THRESHOLD FOR ABOVE IS ETA/SIGMA**2 ',F7.3)
0146      688      $      CONTINUE
0147      688      $      CONTINUE
0148      700      CONTINUE
0149      800      CONTINUE
0150      900      CONTINUE
0151      0152      STOP 0

```

PDP-11 FORTRAN-77 V4.0-1 14:03:46 16-Jul-86
 CLIP2MAA.FTN:2 /F77/MR

```

0001      C SUBROUTINE GETINI,START,DBINC
          C INTERACTIVE INPUT OF PARAMETERS FOR RUN
          C
          C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          C CHARACTER*9 FIELD,BLANK9
          C COMMON /INPUTS/ DEBNOL(3),NSLOTS,K,MN
          C COMMON /SIZE/ NO
          C COMMON /SHARE/ DSNR(3,4)
          C DATA BLANK9/' '
          C WRITE(5,33)
          C FORMAT(1X,BITS/SYMBOL (K) [2]: ',\$')
          C READ(5,3)K
          C IF(K.EQ.0)K=2
          C IF(K.EQ.1)K=2
          C WRITE(5,2)
          C FORMAT(1X, HOW MANY EB/M0? [1]: ',\$')
          C READ(5,3)NO
          C READ(5,3)NO
          C FORMAT(12)
          C FF(NO,EQ.0)NO=1
          C DO 7 IN=1,NO
          C 7 IN=1,NO
          C DO=DSNR(IN,K)
          C WRITE(5,5)IN,DO
          C FORMAT(5,'EB/M0(',12,') ',F9.6,','): ',\$'
          C READ(5,6)FIELD
          C FORMAT(F9.6)
          C IF(FIELD.EQ.BLANK9) THEN
          C   DECODE(9,61,FIELD)DEBNOL(IN)
          C   FORMAT(F9.6)
          C END IF
          C DEBNOL(IN)=DO
          C ELSE
          C   DECODE(9,61,FIELD)DEBNOL(IN)
          C   FORMAT(F9.6)
          C END IF
          C CONTINUE
          C NSLOTS=2400
          C WRITE(5,99)
          C FORMAT(1X, HOW MANY EB/NJ? [51]: ',\$')
          C READ(5,34,ERR=38)NJ
          C FORMAT(13)
          C IF(NJ.EQ.0) NJ=51
          C IF(NJ.LT.0.OR.NJ.GT.51) GOTO 32
          C WRITE(5,41)
          C FORMAT(1X, STARTING VALUE FOR EB/NJ (08) [50]: ',\$')
          C READ(5,42,ERR=40) START
          C IF(START.EQ.0.D0) START=50.D0
          C 42 FORMAT(F6.3)
          C IF(NJ.EQ.1) RETURN
          C 43 WRITE(5,36)
          C FORMAT(1X, DB INCREMENT FOR EB/NJ (-1.0): ',\$')
          C READ(5,37,ERR=35)DBINC
          C FORMAT(F6.3)
          C IF(DBINC.EQ.0.D0) DBINC=-1.0D0
          C RETURN
          C
          C
  
```

Page 5

PDP-11 FORTRAN-77 V4.0-1 14:03:51 16-Jul-86
 CLIP2MAA.FTN:2 /F77/MR

```

0001      SUBROUTINE PSIBE(M,PE,D,IDSUB,MUSED,PRERR,IPSUB,PE00,
          PEAV,IQ)
          S
          C COMPUTE UNCONDITIONAL ERROR PROBABILITY
          C
          C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          C INTEGER JAM(4),JUP(4),JSUB(4)
          C LOGICAL*1 GO,MORE,STORE
          C VIRTUAL PRERR(625),IPSUB(625)
          C VIRTUAL D(625),IDSUB(625)
          C VIRTUAL PEAV(2400)
          C COMMON /SHARE2/ LOW(4),LINC(4)
          C COMMON /DENP/R/ BIGN,AB,BB,LJAM,
          C TAU,TAU2,TAK,TAK2
          C COMMON /PARDEN/ RHOM,RHOT
          C DATA STORE/,TRUE/
          C IF(PEAV(IQ).NE.0.D0) THEN
          C   PE=PEAV(IQ)
          C END IF
          C RETURN
          C
          C 0002
          C 0003
          C 0004
          C 0005
          C 0006
          C 0007
          C 0008
          C 0009
          C 0010
          C 0011
          C 0012
          C 0013
          C 0014
          C 0015
          C 0016
          C 0017
          C 0018
          C 0019
          C 0020
          C 0021
          C 0022
          C 0023
          C 0024
          C 0025
          C 0026
          C 0027
          C 0028
          C 0029
          C 0030
          C 0031
          C 0032
          C 0033
          C 0034
          C 0035
          C 0036
          C 0037
          C 0038
          C 0039
          C 0040
          C 0041
          C 0042
          C 0043
          C 0044
          C 0045
          C 0046
          C 0047
          C 0048
          C 0049
          C 0050
          C
          C THE ALL-ZERO JAMMING EVENT (PE) IS AVAILABLE FROM THE SEARCH FOR
          C THE OPTIMUM THRESHOLD. SO PUT IT INTO THE ARRAY OF SAVED VALUES
          C CALL LOCNM,LM,LUP,JSUB,JSUB1
          C CALL PUTIN(PE00,PRERR,IPSUB,MPS,625,JSUB,KN0F,STORE)
          C IF(KODE.NE.0) STOP 'PRERR FULL'
          C JAM1=1
          C
          C START VECTOR-INDEXED LOOP ON JAMMING EVENTS
          C CALL VLINIT(JAM,LM,M)
          C CONTINUE
          C IF(JAM1.NE.JAM(1)) THEN
          C   UPDATE TEST VALUE FOR NEXT TIME, AND ...
          C   JAM1=JAM(1)
          C END IF
          C
          C BYPASS CONDITIONAL ERROR PROBABILITY CALCULATION IF EVENT
          C PROBABILITY IS ZERO. THIS SAVES MUCH TIME.
          C IF(PIE.EQ.0.D0)GOTO 101
          C
          C SINCE JAMMING PROBABILITIES DEPEND ONLY ON NO. OF CHANNELS
          C HAVING JAM(1) HOPS JAMMED AND NOT THE ARRANGEMENT OF THE
          C CHANNELS, WE CAN SORT THE NON-SIGNAL CHANNELS INTO ASCENDING
          C NUMBERS OF HOPS JAMMED. THIS REDUCES NUMBER OF DISTINCT
          C CONDITIONAL ERROR PROBABILITIES WHICH MUST BE SAVED TO AVOID
          C RECOMPUTING THEM UNNECESSARILY.
          C
          C 0033
          C 0034
          C 0035
          C 0036
          C 0037
          C 0038
          C 0039
          C 0040
          C 0041
          C 0042
          C 0043
          C 0044
          C 0045
          C 0046
          C 0047
          C 0048
          C 0049
          C 0050
          C
  
```

Page 6

PDP-11 FORTRAN-77 V4.0-1 14:03:51 16-Jul-86
CLJPL2M4A.FTN;2 /F77/MR

Page 7

```
0036 IF(M.EQ.2) GOTO 199
0037 DO 110 I=2,M-1
0038 DO 120 J=I+1,M
0039 IF(JSUB(J).LT.JSUB(I)) THEN
0040   JTEMP=JSUB(I)
0041   JSUB(I)=JSUB(J)
0042   JSUB(J)=JTEMP
0043 END IF
0044 CONTINUE
0045 CONTINUE
0046 CONTINUE
0047 C STORE AS ELEMENT OF A SPARSE ARRAY TO SAVE MEMORY
0048 C EVEN THOUGH WE STORE ZEROS, THE SORTING OF SUBSCRIPTS
0049 C CUTS OUT MANY ELEMENTS.
0050 CALL LOC(M,LOW,LUP,JSUB,ISUB)
0051 C TRY TO FIND CONDITIONAL ERROR PROBABILITY IN STORED ARRAY
0052 CALL LOOKUP(PROB,PRERR,IPSUB,NPS,625,ISUB,STORE,NONE)
0053 C IF IT IS NOT THERE, WE MUST COMPUTE IT
0054 IF(NONE) THEN
0055   CALL PSEL(JSUB,M,PROB)
0056   C ... AND SAVE IT FOR POSSIBLE FUTURE RE-USE
0057   CALL PUTIN(PROB,PRERR,IPSUB,MPS,625,ISUB,KODE,STORE)
0058   IF(KODE.NE.0) STOP 2
0059 END IF
0060 C SUM UP UNCONDITIONAL ERROR PROBABILITY
0061 PE=PE+PIE*PROB
0062 C ITERATE THE VECTOR-INDEX LOOP
0063 101 CALL WITER(JAM,LOW,LUP,LINC,M,60)
0064 IF(EO) GOTO 100
0065 RETURN
0066 END
0067
0068
```

PDP-11 FORTRAN-77 V4.0-1 14:03:57 16-Jul-86
CLJPL2M4A.FTN;2 /F77/MR

Page 8

```
0001      SUBROUTINE EVENT(M,JAM,PIE,O,ISUB,MUSED)
0002      C SUBROUTINE TO LOOK UP EVENT PROBABILITY FROM STORED ARRAY
0003      C
0004      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0005      LOGICAL *1 STORE,MORE
0006      DIMENSION JAM(4),LUP(4)
0007      VIRTUAL D(625),ISUB(625)
0008      COMMON /SHARE2/LOM(4),LINC(4)
0009      DATA STORE/.FALSE./
0010      DO 1 I=1,M
0011      LUP(I)=2
0012      CONTINUE
0013      C COMPUTE LINEAR EQUIVALENT SUBSCRIPT FOR JAMMING EVENT
0014      CALL LOC(M,LOW,LUP,JAM,ISUB)
0015      C LOOK UP THE VALUE, GET 0.0 IF NOT THERE
0016      CALL LOOKUP(PIE,O,ISUB,MUSED,G25,ISUB,STORE,NONE)
0017      RETURN
0018 END
```

-5

PDP-11 FORTRAN-77 V4.0-1 14:03:59 16-JUL-86
 CLPL2MAA.FTN:2 /F77/MR

Page 9

PDP-11 FORTRAN-77 V4.0-1 14:03:59 16-JUL-86
 CLPL2MAA.FTN:2 /F77/MR

```

0001      SUBROUTINE GENP(M,N,NSLOTS,GOOD,A,IASUB,C,ICSUB,
0,IDSUB,MUSEC)
0002      C SUBROUTINE TO GENERATE EVENT PROBABILITIES
0003      C
0004      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0005      LOGICAL *1 GO,G02,STORE,MUNE,GO00
0006      DIMENSION LUP2(4),LUP3(4)
0007      DIMENSION IUPA(4)
0008      DIMENSION IUPD(4)
0009      DIMENSION IUP1(4)
0010      DIMENSION IUP0(4)
0011      DATA 1100/100/
0012      DATA IUPA/4*1/
0013      DATA LUP1/4*1/
0014      C STORE= FALSE. => DON'T STORE ZERO ELEMENTS OF SPARSE ARRAY.
0015      C STORE= .TRUE. => STORE ZERO ELEMENTS OF SPARSE ARRAY.
0016      GOOD=.TRUE.
0017      IF (M0,LE,0) THEN
0018          GOOD=.FALSE.
0019          RETURN
0020      END IF
0021      DO 80 L1=1,M
0022          JUPD(L1)=2
0023      80      CONTINUE
0024      C JAMMING PATTERN W/NON-ZERO PROBABILITY ON PER-HOP BASIS
0025      MUSEA=0
0026      C INITIAIZE VECTOR-INDEX LOOP
0027      CALL VLINIT(1,1,LOW,M)
0028      CALL LOCN(1,1,LOW,IUPA,1,1,ISUB)
0029      CALL PRINP(1,M,N,NSLOTS,AIN)
0030      IF (IERR,NE,0) STOP 3
0031      C ITERATE VECTOR-INDEX LOOP
0032      CALL VLITER(1,1,LOW,IUP1),LINE,M,GO)
0033      C COMPUTATION STARTS HERE. FIRST COPY A INTO D.
0034      C SINCE ARRAYS ARE A(0:1,0:1,...,0:1) AND D(0:L,0:L,...,0:L)
0035      C THE COPYING MUST BE DONE ON BASIS OF EQUIVALENT LINEAR
0036      C SUBSCRIPTS RATHER THAN A SIMPLE MOVE OPERATION.
0037      C
0038      C INITIALIZE VECTOR-INDEX LOOP
0039      CALL LOCN(1,1,LOW,IUPD,1,1,ISUB)
0040      CALL LOOKUP(AOUT,D,IDSUB,MUSEA,1100,1,ISUB1,STORE,MUNE)
0041      CALL PUTIN(AOUT,D,IDSUB,MUSEC,625,1,ISUB2,IERR,STORE)
0042      IF (GO) GOTO 99
0043      C ... L-1 CONVOLUTIONS ARE NEEDED ...
0044      DO 9998 L=1,1
0045      C SET UP VECTOR-LOOP UPPER LIMITS FOR THIS CONVOLUTION
0046      DO 125 MN=2,M
0047      LUP2(MN)=1
0048      LUP3(MN)=1+1
0049      MUSEC=0
0050      C LOOK UP ELEMENTS AND PERFORM ONE TERM OF THE CONVOLUTION
0051      CALL LOCN(1,1,LOW,IUPA,1,1,ISUB1)
0052      CALL LOOKUP(AOUT,A,IASUB,MUSEA,1100,1,ISUB1,STORE,MUNE)
0053      DO 21 MN=1,M
0054      L11(MN)=1(MN)+1(MN)
0055      CIN=COUT+AOUT*DOUT
0056      CALL LOCN(1,1,LOW,IUPD,111,1,ISUB3)
0057      CALL LOOKUP(AOUT,D,IDSUB,MUSEC,625,1,ISUB2,STORE,MUNE)
0058      CALL LOOKUP(COUT,C,ICSUB,MUSEC,625,1,ISUB3,STORE,MUNE)
0059      CIN=COUT+AOUT*DOUT
0060      CALL PUTIN(CIN,C,ICSUB,MUSEC,625,1,ISUB3,IERR,STORE)
0061      IF (IERR,NE,0) STOP 4
0062      CALL VLITER(1,1,LOW,IUP2,LINE,M,GO)
0063      IF (GO2) GOTO 97
0064      CALL VLITER(1,1,LOW,IUP3,LINE,M,GO)
0065      IF (GO) GOTO 98
0066      C COPY C TO D IN SORTED ORDER FOR NEXT ITERATION
0067      MUSED=0
0068      CALL VLINIT(1,1,LOW,M)
0069      CONTINUE
0070      CALL LOCN(1,1,LOW,IUPD,1,1,ISUB)
0071      CALL LOOKUP(COUT,D,IDSUB,MUSEB,625,1,ISUB,IERR,STORE)
0072      DIN=COUT
0073      CALL PUTIN(DIN,D,IDSUB,MUSEC,625,1,ISUB,IERR,STORE)
0074      IF (IERR,NE,0) STOP 5
0075      CALL VLITER(1,1,LOW,IUP3,LINE,M,GO)
0076      IF (GO) GOTO 96
0077      CONTINUE
0078      RETURN
0079
    
```

```

PDP-11 FORTRAN-77 V4.0-1 14:04:07 16-Jul-86 Page 11
CLIP12MAA.FTH:2 /F77/MR

0001  SUBROUTINE PUTIN(CIN,C,ICSUB,MUSE,NMAX,K,IERR,STORE)
C THIS SUBROUTINE INSERTS AN ELEMENT INTO A SPARSE ARRAY FOR
C WHICH ONLY THE NON-ZERO ELEMENTS ARE KEPT IN STORAGE IF
C THE SWITCH STORE IS .TRUE.
C
C THE DOUBLE PRECISION VALUE CIN IS STORED AS C(K) WHERE
C THE AUXILIARY ARRAY ICSUB IS USED TO KEEP TRACK OF THE
C SUBSCRIPTS OF THE CORRESPONDING ENTRIES IN C.
C
C USAGE:
C LOGICAL*1 STORE
C DOUBLE PRECISION C,CIN
C VIRTUAL ICSUB(NMAX),C(NMAX)
C CALL PUTIN(CIN,C,ICSUB,MUSE,NMAX,K,IERR,STORE)
C WHERE
C CIN = VALUE OF ELEMENT TO STORE
C C = ARRAY IN WHICH NON-ZERO VALUES ARE ACTUALLY STORED
C ICSUB = AUXILIARY ARRAY FOR ACTUAL SUBSCRIPT VALUES
C MUSE = NUMBER OF ELEMENTS OF C CURRENTLY OCCUPIED
C NMAX = SIZE OF ARRAY C
C IERR = ERROR RETURN CODE, 0 IF NO ERROR OR 1 IF THERE IS
C NO ROOM AVAILABLE IN C
C STORE = .TRUE. TO STORE ZEROES EXPLICITLY ELSE .FALSE.
C NOTE: IF CIN=0 AND THE SUBSCRIPT K IS FOUND IN ICSUB, THEN
C THE ELEMENT IS DELETED BY SHIFTING DOWNWARD ALL
C FOLLOWING ELEMENTS OF THE ARRAY
C
C PROGRAMMER: ROBERT H. FRENCH DATE: 11 JANUARY 1984
C
C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C VIRTUAL ICSUB(NMAX),C(NMAX)
C LOGICAL*1 STORE
C IERR=0
C
C0005  IF(STORE) GOTO 5
C0006  IF(CIN.EQ.0.D0) GOTO 30
C0008  IF(MUSE.EQ.0) GOTO 20
C0009  DO 10 I=1,MUSE
C010   IF(ICSUB(I).NE.K) GOTO 10
C011   C(I)=CIN
C012   RETURN
C013   CONTINUE
C014   IF(MUSE.LT.NMAX) GOTO 20
C015   IERR=1
C016   RETURN
C017   MUSE=MUSE+1
C018   ICSUB(MUSE)=K
C019   C(MUSE)=CIN
C020   RETURN
C021   DO 40 I=1,MUSE
C022   J=I
C023   IF(ICSUB(I).EQ.K) GOTO 50
C024   CONTINUE

```

Page 11

PDP-11 FORTRAN-77 V4.0-1 14:04:07 16-Jul-86 Page 12
CLIP12MAA.FTH:2 /F77/MR

```

0025  RETURN
C REMOVE THE ZERED ELEMENT AND BUMP COUNT OF ENTRIES USED
C
C026  DO 50 I=1,MUSE-1
C027   ICSUB(I)=ICSUB(I+1)
C028   C(I)=C(I+1)
C029   CONTINUE
C030   MUSE=MUSE-1
C031   RETURN
C032   END

PDP-11 FORTRAN-77 V4.0-1 14:04:10 16-Jul-86 Page 13
CLIP12MAA.FTH:2 /F77/MR

0001  SUBROUTINE LOOKUP(COUT,C,ICSUB,N,NMAX,K,STORE,NAME)
C THIS SUBROUTINE RETRIEVES AN ELEMENT OF A SPARSE ARRAY WHICH
C HAS BEEN STORED COMPACTLY BY STOREING ONLY NON-ZERO ELEMENTS.
C
C USAGE:
C VIRTUAL ICSUB(NMAX),C(NMAX)
C LOGICAL*1 STORE, NAME
C DOUBLE PRECISION COUT
C CALL LOOKUP(COUT,C,ICSUB,N,NMAX,K,STORE,NAME)
C WHERE
C COUT = VALUE OF CK(i) (OUTPUT FROM SUBROUTINE)
C C = ARRAY USED TO STORE NON-ZERO ELEMENTS
C ICSUB = AUXILIARY ARRAY TO STORE ACTUAL SUBSCRIPTS
C N = NUMBER OF ELEMENTS OF C CURRENTLY IN USE
C NMAX = SIZE OF C
C K = SUBSCRIPT OF SPARSE ARRAY TO LOOK UP
C STORE = .TRUE. IF ZEROES STORED EXPLICITLY ELSE .FALSE.
C MUSE = .FALSE. IF ZEROES NOT STORED
C .TRUE. IF ZEROES ARE STORED
C
C PROGRAMMER: ROBERT H. FRENCH DATE: 11 JANUARY 1984
C
C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C VIRTUAL ICSUB(NMAX),C(NMAX)
C0002  CONTINUE
C0003  IF(ICSUB(1).NE.10) GOTO 10
C0004  DO 10 I=1,N
C0005  IF(ICSUB(I).NE.K) GOTO 10
C0006  COUT=C(I)
C0007  CONTINUE
C0008  IF(ICSUB(1).NE.10) GOTO 10
C0009  DO 10 I=1,MUSE
C010   IF(ICSUB(I).NE.K) GOTO 10
C011   COUT=C(I)
C012   RETURN
C013   CONTINUE
C014   IF(MUSE.LT.NMAX) GOTO 20
C015   IERR=1
C016   RETURN
C017   MUSE=MUSE+1
C018   ICSUB(MUSE)=K
C019   C(MUSE)=CIN
C020   RETURN
C021   DO 40 I=1,MUSE
C022   J=I
C023   IF(ICSUB(I).EQ.K) GOTO 50
C024   CONTINUE

```

PDP-11 FORTRAN-77 V4.0-1 14:04:07 16-Jul-86 Page 12
CLIP12MAA.FTH:2 /F77/MR

POP-11 FORTRAN-77 V4.0-1 14:04:12 16-Jul-86
CLPL2MA.FTN;2 /F77/BR

Page 14

0001 SUBROUTINE LOCN(NDIM, ILON, IUP, ISUB, LINEAR)
C THIS SUBROUTINE COMPUTES THE EQUIVALENT LINEAR SUBSCRIPT FOR
C A MULTIDIMENSIONAL ARRAY OF NDIM DIMENSIONS
C IF THE ARRAY A IS DEFINED AS
C DIMENSION A(ILON(1):IUP(1),...,ILON(NDIM):IUP(NDIM))
C AND ISUB(1),...,ISUB(NDIM) IS A SET OF SUBSCRIPTS FOR A,
C THEN THIS SUBROUTINE RETURNS IN LINEAR THE OFFSET FROM THE
C ORIGIN OF A TO THE ELEMENT A(ISUB(1),...,ISUB(NDIM)). ASSUMING
C THE FIRST SUBSCRIPT VARIES MOST RAPIDLY.
C USAGE:
C DIMENSION ILON(NDIM),IUP(NDIM),ISUB(NDIM)
C DATA ILON/Ilower limits of defined subscripts of array/
C DATA IUP/upper limits of defined subscripts of array/
C C...SET ISUB TO DESIRED SEQUENCE OF SUBSCRIPTS...
C WHERE
C NDIM = NUMBER OF DIMENSIONS THE ARRAY HAS
C ILON = ARRAY OF LOWER SUBSCRIPT BOUNDS
C IUP = ARRAY OF UPPER SUBSCRIPT BOUNDS
C ISUB = ARRAY CONTAINING SUBSCRIPT FOR WHICH LOCATION IS
C TO BE COMPUTED
C LINEAR = RETURNED VALUE OF OFFSET INTO ARRAY IN MEMORY

C PROGRAMMER: ROBERT H. FRENCH

C DATE: 11 JANUARY 1984

0002 DIMENSION ILON(NDIM),IUP(NDIM),ISUB(NDIM)
0003 LINEAR=0
0004 DO 10 I=1,NDIM-1
0005 J=NDIM-I+1
0006 LINEAR=(LINEAR+(ISUB(J)-ILON(J)))*(IUP(J-1)-ILON(J-1)+1)
0007 10 CONTINUE
0008 LINEAR=LINEAR+ISUB(1)-ILON(1)
0009 RETURN
0010 END

POP-11 FORTRAN-77 V4.0-1 14:04:14 16-Jul-86
CLPL2MA.FTN;2 /F77/BR

Page 15

0001 SUBROUTINE VLINIT(LVEC,LLON,LMAX)
C THIS SUBROUTINE INITIALIZES A "VECTOR DO-LOOP" STRUCTURE
C DEFINED BY THE FOLLOWING PSEUDO-FORTRAN CODE:
C DO 100 LVEC(1)=LLON(1),LUP(1),LINC(1)
C DO 100 LVEC(2)=LLON(2),LUP(2),LINC(2)
C :
C DO 100 LVEC(LMAX)=LLON(LMAX),LUP(LMAX),LINC(LMAX)
C :
C (STATEMENTS IN RANGE OF LOOP)
C 100 CONTINUE
C THE COMPANION ROUTINE VLITER HANDLES THE LOOP CONTROL AT THE
C CONTINUE STATEMENT IN THE ABOVE STRUCTURE
C USAGE:
C LOGICAL *1 GO
C DIMENSION LVEC(LMAX),LLON(LMAX),LUP(LMAX),LINC(LMAX)
C (INITIALIZE ARRAY LLON TO STARTING VALUES OF THE NESTED LOOPS)
C (INITIALIZE ARRAY LUP TO STOPPING VALUES OF THE NESTED LOOPS)
C (INITIALIZE ARRAY LINC TO INCREMENTS OF THE LOOPS)
C CALL VLINIT(LVEC,LLON,LMAX)
C 100 CONTINUE
C : (STATEMENTS IN RANGE OF LOOPS)
C CALL VLITER(LVEC,LLON,LUP,LINC,LMAX,GO)
C WHERE
C LVEC = ARRAY FOR STORAGE OF LOOP INDICES. LVEC(1) IS THE
C OUTER-MOST LOOP; LVEC(LMAX) THE INNER-MOST LOOP.
C LLON = ARRAY FOR STORAGE OF LOOP STARTING VALUES. IN SAME
C SEQUENCE AS LVEC
C LUP = ARRAY FOR STORAGE OF LOOP FINAL VALUES. IN SAME
C SEQUENCE AS LVEC
C LINC = ARRAY FOR STORAGE OF LOOP INCREMENTS. IN SAME
C SEQUENCE AS LVEC
C LMAX = NUMBER OF LOOPS NESTED
C GO = LOGICAL VARIABLE. TRUE. IF JUMP BACK TO BEGINNING OF
C STATEMENTS IN THE RANGE OF THE LOOP SHOULD OCCUR,
C FALSE. OTHERWISE (I.E. OUTER-MOST LOOP TERMINATED)
C PROGRAMMER: ROBERT H. FRENCH DATE: 11 JANUARY 1984
C
C DIMENSION LVEC(LMAX),LLON(LMAX)
C DO 1 N=1,LMAX
C 0003 LVEC(N)=LLON(N)
C 0004 CONTINUE
C 0005 1
C 0006 RETURN
C 0007 END

```

0001      SUBROUTINE VLITER(LVEC,LLOW,LUP,LINC,IMAX,GO)
0002
0003      C LOOP ITERATION LOGIC FOR A "VECTOR DO-LOOP"
0004      C SEE DETAILED COMMENTS IN SUBROUTINE VLIMIT FOR USAGE AND
0005      C PARAMETER DEFINITIONS
0006      C
0007      C PROGRAMMER: ROBERT H. FRENCH
0008      C DATE: 11 JANUARY 1984
0009
0010      LOGICAL*1 GO
0011      DIMENSION LVEC(IMAX),LLOW(IMAX),LUP(IMAX),LINC(IMAX)
0012      GO=.TRUE.
0013      DO 100 MDX=1,IMAX
0014      NSUB=LMAX+1-MDX
0015      LVEC(NSUB)=LVEC(NSUB)+LINC(NSUB)
0016      IF((LINC(NSUB).GE.0).AND.LVEC(NSUB).LE.LUP(NSUB)))
0017      .OR.(LINC(NSUB).LT.0.AND.LVEC(NSUB).GE.LUP(NSUB))) RETURN
0018      LVEC(NSUB)=LLOW(NSUB)
0019      CONTINUE
0020      GO=.FALSE.
0021      RETURN
0022      END
0023
0024
0025
0026

```

```

0001      C THIS SUBROUTINE COMPUTES THE EVENT PROBABILITY FOR ALL
0002      C POSSIBLE JAMMING PATTERNS WITH NON-ZERO PROBABILITY FOR
0003      C L=1 HOP/SYMBOL FOR RMF SX/FH IN PBNAJ
0004
0005      C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0006      C DIMENSION I(4)
0007      C
0008      C IF THIS IS AN IMPOSSIBLE CASE, RETURN WITH RESULT = 0.0
0009      C IF(KJAM.GT.MINO(KQ,KM)) RETURN
0010      KJAM=KJAM-1
0011      LMAX=KM-KJAM-1
0012      JPMAX=KM-1
0013      IMAX=MAJO(KPMax,LPMax,JPMax)
0014      PROD=1.00
0015      Q=KQ
0016      DIFFQ=KN-KQ
0017      EN=IN
0018      DO 100 LLOOP=0,IMAX
0019      F=LLOOP
0020      IF(LLOOP.LE.KPMax) PROD=PROD*(0-F)
0021      IF(LLOOP.LE.JPMax) PROD=PROD*(EN-F)
0022      IF(LLOOP.LE.LPMax) PROD=PROD*(DIFFQ-F)
0023      100  CONTINUE
0024      AIM=PROD
0025
0026      RETURN
END

```

```

0001      C BLOCK DATA
0002      C INITIALIZE SHARED CONSTANTS
0003      C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003      C      COMMON /SHARE/ DSRR(1:4)
0004      C      COMMON /SHARE/ LM(14),LINC(4)
0005      C      COMMON /NTS/ X(8),M(8)

C WEIGHTS AND ABSISSAS FOR 16-POINT GAUSSIAN QUADRATURE
0006      DATA X/ 0.09501250983763744018500,
0006      S   0.28160355907955891123000,
0006      S   0.45801677765722738634200,
0006      S   0.61787624440264374844700,
0006      S   0.7554044008355003033889500,
0006      S   0.86563120238783174388000,
0006      S   0.94457502307323257607800,
0006      S   0.98940093499164993259600 /
0007      DATA W/ 0.18945061045506849628500,
0007      S   0.18260341504402358886700,
0007      S   0.16915651939500253818900,
0007      S   0.14959598881657673206100,
0007      S   0.12462897125533387205200,
0007      S   0.0951585116829278481000,
0007      S   0.06225352393864789286300,
0007      S   0.02715245941175409485200 /
C DEFAULT LISTS FOR INTERACTIVE PARAMETER INPUTS
C ARE SHARED WITH LARGE WORKING STORAGE ARRAYS SINCE THEY
C MAY BE DESTROYED ONCE THE INPUT PARAMETERS ARE SET UP
0008      DATA DSRR /13.3524700, 12.313300, 10.9444300,
0008      S   10.60657200, 9.628400, 8.3524800,
0008      S   9.0940100, 8.169000, 6.97199500,
0008      S   8.0783500, 7.199600, 6.06964600/
C FREQUENTLY NEEDED CONSTANT ARRAYS AND SCALARS
0009      DATA LM/4*0,LINC/4*1/
0010      END

```

```

0001      SUBROUTINE FSEL(JSUB,MN,PROB)
0001      C RANDOM WFSK/FH IN PARTIAL BAND NOISE JAMMING.
0001      C GIVEN A JAMMING EVENT, WITH CLIPPER RECEIVER
0001      C JSUB - JAMMING EVENT VECTOR
0001      C MN - ALPHABET SIZE
0001      C PROB - RESULTING CONDITIONAL ERROR PROBABILITY
0001      C
0002      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0002      DIMENSION JSUB(MN), WORK(30), STACK(30), HEAP(30)
0002      EXTERNAL FG16, PGRAFD
0002      INTEGER NCHAN(0:3)
0002      COMMON /JAMCNTR/ NCHAN
0002      COMMON /DEPAR/ BIK, AAB, BAS, LJMM,
0002      TAU, TAU2, TAUK, TAUK2
0002      S
0003      COMMON /PARDEN/ RHOM, RHOT
0003      COMMON /QUES/ QD, QI
0003      COMMON /SCJAM/ JAMSC
0003      KSUB=0
0003      DO 6 I=1,MN
0003      KSUB=KSUB+JSUB(I)
0004      6 CONTINUE
0004      C SET UP VALUES WHICH WILL REMAIN IF THIS IS THE NOTHING-JAMMED CASE
0004      Q=Q(2.00*DQR(0.500+RHOM),DQR(1.00+TAU))
0005      Q1=1.00
0005      0110
0005      0011
0005      0012
0005      0013
0005      0014
0005      0015
0005      0016
0005      0017
0005      0018
0005      0019
0005      0020
0005      0021
0005      0022
0005      0023
0005      0024
0005      0025
0005      0026
0005      C COUNT NUMBER OF MONOSIGNAL CHANNELS WITH LM HOPS JAMMED
0005      C
0006      0027      DO 10 I=0,2
0006      0028      NCHAN(I)=0
0006      0029      CONTINUE
0006      10      DO 11 I=2,MN
0006      0030      KSUB=JSUB(I)
0006      0031      NCHAN(KSUB)=NCHAN(KSUB)+1
0006      0032      11      CONTINUE
0006      0033      0034      JAMSC=JSUB(1)
0006      0035      C DO THE CONTINUOUS PART OF THE DENSITY IN SECTIONS
0006      PCONT=0.00

```

PPP-11 FORTRAN-77 V4.0-1
CLIP12MA.FTN;2 /F77/MR

Page 20

PPP-11 FORTRAN-77 V4.0-1 14:04:23 16-Jul-86
CLIP12MA.FTN;2 /F77/MR

```
0036      DO 13 ISECT=1,2
0037      XL=(ISECT-1)*TAU
0038      XU=ISECT*TAU
0039      CALL ADQUAD(XL,XU,CRUNK,D616,PGRAND,1,D-8,WORK,
0040           STACK,HEAP,30,KODE)
0041      S          CALL TEST2(KODE,10)
0042      PCONT=PCONT+CRUNK
0043      13  CONTINUE
0044      C  DO THE TIE PART OF THE DENSITY
0045      CALL TIES(JSUB,MN,PTIE)
0046      C  PUT THEM TOGETHER
0047      PROB=1.00-PCONT-PTIE
0048      RETURN
0049      END
```

```
I-11
0001      DOUBLE PRECISION FUNCTION PGRAND(BETA)
0002      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003      INTEGER NCHAN(0:3)
0004      COMMON /SCJAM/ JAMSC
0005      COMMON /JAMENT/ NCHAR
0006      COMMON /DEMPAR/ BISK, AAB, BAB, LJAM
0007      COMMON /PARDEN/ RHON, RHOI
0008      COMMON /QUES/ QD, QI
0009      PROD=1.00
0010      DO 10 I=0,2
0011      IF(NCHAN(I).NE.0) THEN
0012      LJAM=I
0013      X=GL(BETA)
0014      PROD=PROD*DX1(I,X,NCHAN(I))
0015      END IF
0016      10  CONTINUE
0017      LJAM=JAMSC
0018      Y=PZ1(BETA)
0019      PGRAND=Y*PROD
0020      RETURN
0021
```

Page 21

PPP-11 FORTRAN-77 V4.0-1 14:04:27 16-Jul-86
CLIP12MA.FTN;2 /F77/MR

Page 22

```
0001      SUBROUTINE ADQUAD(XL,XU,Y,QR,F,TOL,WORK,STACK,HEAP,N,KODE)
0002
0003      C  ADAPTIVE QUADRATURE ALGORITHM
0004      C  XL - LOWER LIMIT OF INTEGRAL (IN)
0005      C  XU - UPPER LIMIT OF INTEGRAL (IN)
0006      C  Y   - VALUE OF INTEGRAL (OUT)
0007      C  QR  - NAME OF A QUADRATURE RULE SUBROUTINE (IN)
0008      C  WITH CALLING SEQUENCE
0009      C  CALL QR(XL,XU,F,Y)
0010      C  F   - NAME OF FUNCTION TO BE INTEGRATED (IN)
0011      C  TOL - ERROR TOLERANCE FOR FINAL ANSWER (IN)
0012      C  WORK - WORK ARRAY OF SIZE N (IN)
0013      C  STACK- SECOND WORK ARRAY OF SIZE N, MUST NOT BE
0014      C  HEAP- THIRD WORK ARRAY, SIZE N, DISTINCT FROM WORK AND STACK
0015      C  SAME ARRAY AS WORK (IN)
0016      C  N   - SIZE OF WORK AND STACK; MAX. NO. OF BISECTIONS (IN)
0017      C  KODE - ERROR INDICATOR (OUT)
0018      C  0 -- NO ERROR
0019      C  1 -- WORK ARRAYS TOO SMALL
0020      C  2 -- EPS DIVIDED TO ZERO, EITHER ASKING FOR TOO
0021      C    TIGHT A TOLERANCE OR ROUND-OFF PREVENTS
0022      C    ATTAINING REQUIRED ACCURACY
0023      C  R. H. FRENCH, 14 AUGUST 1984
0024
0025      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0026      EXTERNAL F
0027      DIMENSION WORK(N),STACK(N),HEAP(M)
0028      KODE=0
0029      Y=0.00
0030      WORK(1)=XU
0031      CALL QR(XL,XU,F,T)
0032      HEAD(1)=T
0033      A=XL
0034      NPITS=1
0035      EPS=TOL
0036      STACK(1)=EPS
0037      10  B=WORK(NPITS)
0038      XM=(A+B)*0.500
0039      CALL QR(A,XM,F,P1)
0040      IF(DABS(T-P1).LE.EPS) GOTO 20
0041      C  SPLIT IT
0042      NPITS=NPITS+1
0043      IF(NPITS.GT.N) THEN
0044      KODE=1
0045      RETURN
0046      END IF
0047      WORK(NPITS)=XM
0048      HEAP(NPITS)=P2
0049      T=P1
0050      EPS=EPS/2.00
0051
```

```

PPDP-11 FORTRAN-77 V4.0.1 14:04:30
CLIP12MAA.FTN;2 /F77/MR 16-Jul-86

      IF(EPS.EQ.0.00) THEN
        KODE=2
        RETURN
      END IF
      STACK(NPTS)=EPS
      GOTO 10
    C FINISHED A PIECE
    20   Y=+P1+P2
        EPSSTACK(NPTS)
        T=MEAP(NPTS)
        NPTS=NPTS-1
        A=B
        IF(NPTS.EQ.0) RETURN
      GOTO 10
    END

```

```

0001      SUBROUTINE ADQUA2(XL,XU,Y,QR,F,TOL,WORK,STACK,HEAP,N,KODE
0002
0003      C ADAPTIVE QUADRATURE ALGORITHM
0004      C XL - LOWER LIMIT OF INTEGRAL (IN)
0005      C XU - UPPER LIMIT OF INTEGRAL (IN)
0006      C Y - VALUE OF INTEGRAL (OUT)
0007      C QR - NAME OF A QUADRATURE RULE SUBROUTINE (IN)
0008      C WITH CALLING SEQUENCE
0009      C      CALL QR(XL,XU,Z,Y)
0010      C F - NAME OF FUNCTION TO BE INTEGRATED (IN)
0011      C TOL - ERROR TOLERANCE FOR FINAL ANSWER (IN)
0012      C WORK - WORK ARRAY OF SIZE N (IN)
0013      C STACK - SECOND WORK ARRAY OF SIZE N, MUST NOT BE
0014      C      WORK - THIRD WORK ARRAY, SIZE N, DISTINCT FROM WORK AND STACK
0015      C      SAME AS WORK (IN)
0016      C      N - SIZE OF WORK AND STACK; MAX. NO. OF BISECTIONS (IN)
0017      C KODE - ERROR INDICATOR (OUT)
0018      C      0 --- NO ERROR
0019      C      1 -- WORK ARRAYS TOO SMALL
0020      C      2 -- EPS DIVIDED TO ZERO, EITHER ASKING FOR TOO
0021      C      TIGHT A TOLERANCE OR ROUND-OFF PREVENTS
0022      C      ATTAINING REQUIRED ACCURACY
0023
0024      C R. H. FRENCH, 14 AUGUST 1984
0025
0026      IMPLICIT DOUBLE PRECISION(A,H,O,Z)
0027      EXTERNAL F
0028      DIMENSION WORK(N),STACK(N),HEAP(N)
0029      KODE=0
0030      Y=0.0D0
0031      WORK(1)=XU
0032      CALL QR(XL,XU,F,T)
0033      HEAP(1)=T
0034      A=XL
0035      NPTS=1
0036      EPS=TOL
0037      STACK(1)=EPS
0038      B=WORK(NPTS)
0039      XN=(A+B)/0.500D0
0040      CALL QR(A,XN,F,P1)
0041      CALL QR(XN,B,F,P2)
0042      IF(DABS(T-P1-P2)>LE.EPS) GOTO 20
0043      C SPLIT IT
0044      NPTS=NPTS+1
0045      IF(NPTS.GT.N) THEN
0046          KODE=1
0047          RETURN
0048      END IF
0049      WORK(NPTS)=XN
0050      HEAP(NPTS)=P2
0051      T=P1
0052      EPS=EPS/2.D0
0053

```

```

POP-11 FORTRAN-77 V4.0-1   14:04:33   16-Jul-86   Page 25
CLPL2MAA.FTN;2   /F77/WR

0028 IF(EPS.EQ.0.D0) THEN
0029   KONE=2
0030   RETURN
0031 END IF
0032 STACK(NPTS)=EPS
0033 GOTO 10
0034 C FINISHED A PIECE
0035   20 Y=Y+P1+P2
0036   EPS=STACK(NPTS)
0037   T=HEAP(NPTS)
0038   NPTS=NPTS-1
0039   A=B
0040   IF(NPTS.EQ.0) RETURN
0041   GOTO 10
0042 END

```

```

POP-11 FORTRAN-77 V4.0-1   14:04:36   16-Jul-86   Page 26
CLPL2MAA.FTN;2   /F77/WR

0001 SUBROUTINE D616(A,B,F,ANSWER)
0002 C 16-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL
0003 C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4
0004 C
0005 C R. H. FRENCH, 28 FEBRUARY 1986
0006 C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0007 COMMON /MTS/ X(8),M(8)
0008 ANSWER=0.00
0009 BPA02=(B-A)/2.00
0010 BPA02=(B-A)/2.00
0011 DO 10 I=1,8
0012   C=X(I)*BPA02
0013   Y1=BPA02+C
0014   Y2=BPA02-C
0015 CONTINUE
0016 ANSWER=ANSWER+W(I)*(F(Y1)+F(Y2))
0017 10 ANSWER=ANSWER+BPA02
0018 RETURN
0019 END

POP-11 FORTRAN-77 V4.0-1   14:04:38   16-Jul-86   Page 27
CLPL2MAA.FTN;2   /F77/WR

0001 SUBROUTINE D617(A,B,F,ANSWER)
0002 C 16-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL
0003 C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4
0004 C
0005 C R. H. FRENCH, 28 FEBRUARY 1986
0006 C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0007 COMMON /MTS/ X(8),M(8)
0008 ANSWER=0.00
0009 BPA02=(B-A)/2.00
0010 BPA02=(B+A)/2.00
0011 DO 10 I=1,8
0012   C=X(I)*BPA02
0013   Y1=BPA02+C
0014   Y2=BPA02-C
0015 CONTINUE
0016 ANSWER=ANSWER+W(I)*(F(Y1)+F(Y2))
0017 10 ANSWER=ANSWER+BPA02
0018 RETURN
0019 END

```

PDP-11 FORTRAN-77 V4.0-1 14:04:40 16-Jul-86
 CLIP12MA.FTN;2 /F77/MR

0001 SUBROUTINE TEST(ID)
 C TEST RETURN CODE FROM BESSEL FUNCTION
 C
 0002 IMPLICIT DOUBLE PRECISION(A-I,O-Z)
 COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, I01, I02
 IF(KODE.EQ.0) RETURN
 WRITE(5,1) KODE, ID
 0003 1 FORMAT('ADAPTIVE INTEGRATOR CODE = ',I2,
 0004 ' FROM CALL NUMBER ',I5)
 0005 STOP 'FATAL ERROR.'
 END

0006

0007

0008

Page 28

PDP-11 FORTRAN-77 V4.0-1 14:04:41 16-Jul-86
 CLIP12MA.FTN;2 /F77/MR

Page 29

0001 C SUBROUTINE TEST2(KODE, ID)
 C TEST RETURN CODE FROM ADQUAD/ADQUA2
 C
 0002 IF(KODE.EQ.0) RETURN
 WRITE(5,1) KODE, ID
 0003 1 FORMAT('ADAPTIVE INTEGRATOR CODE = ',I2,
 0004 ' FROM CALL NUMBER ',I5)
 0005 STOP 'FATAL ERROR.'
 END

0006

```

0001      C DOUBLE PRECISION FUNCTION P21(Y)
          C SIGNAL CHANNEL P.D.F. WITH CHANGE OF VARIABLE Y=AX
          C
          C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
          C DIMENSION WORK(30), STACK(30), HEAP(30)
          C LOGICAL*1 REG1, REG2
          C EXTERNAL DBXY1, F20, F21, F22
          C COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
          C COMMON /INPAR/ BICK, AAB, BAR, LJAM,
          C TAU, TAU2, TAUK, TAUW2
          C COMMON /PARDEN/ RHOM, RHOT
          C COMMON /QUES/ QO, TO
          C COMMON /XCON/ XCON
          C COMMON /OUTER/ XXX, XXXX
          C XXX=Y
          C IF(LJAM.GE.1) THEN
          C   XXX=Y/BICK
          C   YK=XXX
          C   YTK=(Y-TAU)/BICK
          C   YTK2=(Y-TAU2)/BICK
          C   END IF
          C   REG1=Y GE. 0. DO .AND. Y.LT.TAU
          C   REG2=Y GE.TAU .AND. Y.LT.TAU2
          C   -15
          C   TWO HOPS PER SYMBOL
          C
          C 2000 6010 (2100, 2200, 2300), LJAM+1
          C
          C NO HOPS JAMMED
          C
          C 2100  IF(REG1) THEN
          C    BARG1=DSQRT(8.00*RHOM*Y)
          C    101=1
          C    CALL DBBT1(2100)
          C    P21=0.500*DSQRT(2.00*Y/RHOM)*DEXP(BARG1-Y-2.00*RHOM)*B1
          C    ELSE IF(REG2) THEN
          C      BARG1=DSQRT(4.00*RHOM*(Y-TAU))
          C      101=0
          C      CALL DBBT1(2101)
          C      PART=2.00*DEXP(BARG1-Y-TAU-RHOM)*B1
          C      XCON=Y+2.00*RHOM
          C      CALL ADQUA2(Y-TAU, TAU, ANSWER, DGXY1, F20, 1, 0-9, WORK, STACK,
          C      HEAP, 30, KOD)
          C
          C 0021  CALL TEST2(KOD, 2100)
          C      P21=PART+ANSWER/(BICK*BICK)
          C
          C 0022  ELSE
          C      P21=0.00
          C
          C 0023  END IF
          C
          C 0024  CALL TEST2(KOD, 2100)
          C      P21=PART+ANSWER
          C
          C 0025  ELSE
          C      P21=0.00
          C
          C 0026  END IF
          C
          C 0027  END IF
          C
          C 0028  END IF
          C
          C 0029  END IF
          C
          C 0030  END IF
          C
          C 0031  END IF
          C
          C 0032  END IF
          C
          C 0033  END IF
          C
          C 0034  END IF
          C
          C 0035  END IF
          C
          C 0036  END IF
          C
          C 0037  END IF
          C
          C 0038  END IF
          C
          C 0039  END IF
          C
          C 0040  ONE HOP JAMMED
          C
          C 0041  IF(REG1) THEN
          C    XCON=YK*RHOM+RHOT
          C    CALL ADQUA2(0, DO_Y, ANSWER, DGXY1, F21, 1, 0-9, WORK, STACK,
          C    HEAP, 30, KOD)
          C
          C 0042  $ CALL TEST2(KOD, 2200)
          C    P21=ANSWER/BICK
          C
          C 0043  ELSE IF(REG2) THEN
          C    BARG1=DSQRT(4.00*RHOT*YTK)
          C    101=0
          C    CALL DBBT1(2201)
          C    PART=Q0*DEXP(BARG1-YK-TAUK-RHOT)*B1/BICK
          C    BARG1=DSQRT(4.00*RHOM*(Y-TAU))
          C
          C 0044  CALL DBBT1(2202)
          C    PART=PART+Q1*DEXP(BARG1-Y-TAU-RHOM)*B1
          C    XCON=YK*RHOM+RHOT
          C
          C 0045  CALL ADQUA2(Y-TAU, TAU, ANSWER, DGXY1, F21, 1, 0-9, WORK, STACK,
          C    HEAP, 30, KOD)
          C
          C 0046  P21=PART+ANSWER/BICK
          C
          C 0047  END IF
          C
          C 0048  END IF
          C
          C 0049  END IF
          C
          C 0050  END IF
          C
          C 0051  END IF
          C
          C 0052  END IF
          C
          C 0053  END IF
          C
          C 0054  END IF
          C
          C 0055  END IF
          C
          C 0056  END IF
          C
          C 0057  END IF
          C
          C 0058  END IF
          C
          C 0059  END IF
          C
          C 0060  END IF
          C
          C 0061  END IF
          C
          C 0062  TWO HOPS JAMMED
          C
          C 0063  IF(REG1) THEN
          C    BARG1=DSQRT(8.00*RHOT*YK)
          C    101=1
          C    CALL DBBT1(2300)
          C    P21=0.500*DSQRT(2.00*Y/RHOM)*DEXP(BARG1-YK-2.00*RHOT)
          C
          C 0064  ELSE IF(REG2) THEN
          C    BARG1=DSQRT(4.00*RHOT*YTK)
          C    101=0
          C    CALL DBBT1(2301)
          C    PART=Q1*DEXP(BARG1-YTK-RHOT)*B1/BICK
          C
          C 0065  XCON=YK+2.00*RHOM
          C    CALL ADQUA2(Y-TAU, TAU, ANSWER, DGXY1, F22, 1, 0-9, WORK, STACK,
          C    HEAP, 30, KOD)
          C
          C 0066  CALL TEST2(KOD, 2300)
          C    P21=PART+ANSWER/(BICK*BICK)
          C
          C 0067  ELSE
          C    P21=0.00
          C
          C 0068  END IF
          C
          C 0069  END IF
          C
          C 0070  END IF
          C
          C 0071  END IF
          C
          C 0072  END IF
          C
          C 0073  END IF
          C
          C 0074  END IF
          C
          C 0075  END IF
          C
          C 0076  END IF
          C
          C 0077  END IF
          C
          C 0078  END IF
          C
          C 0079  CONTINUE
          C
          C 0080  RETURN
          C
          C 0081  END

```

```

PDP-11 FORTRAN-77 V4.0-1 14:04:50 16-JUL-86
CLTPL2M4A.FTN.2 /F77/NR

0001      C DOUBLE PRECISION FUNCTION GL(Y)
C
C   C NON-SIGNAL CHANNEL CUMULATIVE DISTRIBUTION
C   C WITH CHANGE OF VARIABLE Y-AX
C
C   C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C   LOGICAL T1, REG1, REG2, REG3
C   COMMON /DEMPAR/ BICK, AAB, BAB, LJAM,
C   TAU, TAU2, TAUK, TAUK2
C
C   S
C   REG1=Y.GE.0.00 .AND. Y.LT.TAU
C   REG2=Y.GE.TAU .AND. Y.LT.TAU2
C   IF(LJAM.GT.0) THEN
C     YK=Y.BICK
C   YTK=(Y.TAU)/BICK
C   YTR2=(Y-TAU2)/BICK
C   END IF
C
C   C TWO HOPS PER SYMBOL
C   2000  GOTO (2100, 2200, 2300), LJAM+1
C
C   C NO HOPS JAMMED
C
C   2100  IF(REG1) THEN
C         GL=1.00-(1.00+Y)*DEXP(-Y)
C       ELSE IF(REG2) THEN
C         GL=1.00-(1.00+TAU2-Y)*DEXP(-Y)
C       ELSE IF(Y.GE.TAU2) THEN
C         GL=1.00
C       ELSE
C         GL=0.00
C       END IF
C   6070  9000
C
C   C ONE HOP JAMMED
C
C   2200  IF(REG1) THEN
C         GL=1.00-AAB*DEXP(-YK)+BAB*DEXP(-Y)
C       ELSE IF(REG2) THEN
C         GL=1.00-AAB*DEXP(TAU-TAUK-Y)+BAB*DEXP
C       ELSE IF(Y.GE.TAU2) THEN
C         GL=1.00
C       ELSE
C         GL=0.00
C       END IF
C   6070  9000
C
C   C TWO HOPS JAMMED
C
C   2300  IF(REG1) THEN
C         GL=1.00-(1.00+YK)*DEXP(-YK)

```

```

PDP-11 FORTRAN-77 V4.0-1      14:04:50      16-Jul-86
CL1PL2MAA.FTN;2          /F77/MR      Page 33

0035    ELSE IF (REG2) THEN
0036        GL=1.00-(1.00+TAU2-YK)*DEXP(-YK)
0037        ELSE IF (Y.GE.TAU2) THEN
0038            GL=1.00
0039        ELSE
0040            GL=0.00
0041        END IF
0042        GOTO 9000
0043        CONTINUE
0044        RETURN
0045    END

PDP-11 FORTRAN-77 V4.0-1      14:04:54      16-Jul-86
CL1PL2MAA.FTN;2          /F77/MR      Page 34

0001    C   DOUBLE PRECISION FUNCTION F20(U)
C   IMTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=2, LJAM=1
0002    IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003    COMMON /DNPAR/ B1SK, AAB, BAR, LJAM,
0004    $                TAU, TAU2, TAUK, TAUK2
0005    COMMON /PARDEN/ RHOIN, RHOT
0006    COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, I01, I02
0007    COMMON /QUES/ Q0, Q1
0008    COMMON /OUTER/ XXX, XXX
0009    BARG1=DSQRT(4.D0+RHOIN*U)
0010    BARG2=DSQRT(4.D0+RHOIN*(XXX-U))
0011    I01=0
0012    I02=0
0013    CALL BPR00(2110)
0014    F20=DEXP(BARG1+BARG2-XCON)*B1*82
0015    RETURN
0016

```

```

0001      C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=2, LJMM=1
          C
          C DOUBLE PRECISION FUNCTION F21(U)
          C
          C IMPLICIT DOUBLE PRECISION(A-H,0-Z)
          C COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
          C COMMON /DEMPAR/ BIGK, AAB, BAR, LJMM,
          C                   TAU, TAU2, TAUK, TAUK2
          $ COMMON /PARDEN/ RHON, RHOT
          COMMON /QUES/ QO, Q1
          COMMON /XCOM/ XCON
          COMMON /OUTER/ XXX, XXXX
          BARG1=DSQRT(4.D0*RHON*U)
          BARG2=DSQRT(4.D0*RHOT*(XXXK-U/BIGK))
          101=0
          102=0
          CALL BPROD(2210)
          F21=DEXP(BARG1+BARG2-XCON-U+U/BIGK)*B1*B2
          RETURN
          END

```

Page 36
PDP-11 FORTRAN-77 V4.0-1 /F77/MR 14:04:57 16-Jul-86

I - 17 SUBROUTINE BPROD(IDENT)

```

          C COMPUTE TWO BESSSEL FUNCTIONS, ARGUMENTS AND RESULTS IN COMMON
          C
          C IMPLICIT DOUBLE PRECISION(A-H,0-Z)
          C COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
          C CALL DXB1(IDENT)
          C CALL DXBESI(BARG2,102,B2,KODE)
          C CALL DXBESI(BARG1,101,B1,KODE)
          C CALL TEST(IDENT+1)
          C RETURN
          END

```

```

0001      C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=2, LJMM=2
          C
          C DOUBLE PRECISION FUNCTION F22(U)
          C
          C IMPLICIT DOUBLE PRECISION(A-H,0-Z)
          C COMMON /DEMPAR/ BIGK, AAB, BAR, LJMM,
          C                   TAU, TAU2, TAUK, TAUK2
          $ COMMON /PARDEN/ RHON, RHOT
          COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
          COMMON /XCOM/ XCON
          COMMON /OUTER/ XXX, XXXX
          BARG1=DSQRT(4.D0*RHOT*U/BIGK)
          BARG2=DSQRT(4.D0*RHOT*(XXXK-U/BIGK))
          101=0
          102=0
          CALL BPROD(2310)
          F22=DEXP(BARG1+BARG2-XCON)
          RETURN
          END

```

Page 38
PDP-11 FORTRAN-77 V4.0-1 /F77/MR 14:05:00 16-Jul-86

I - 17 SUBROUTINE DXBT(ID)

```

0001      C CALL DXBESI AND TEST RETURN CODE
          C
          C IMPLICIT DOUBLE PRECISION(A-H,0-Z)
          C COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
          C CALL DXBESI(BARG1,101,B1,KODE)
          C CALL TEST(ID)
          C RETURN
          END

```

```

POP-11 FORTRAN-77 V4.0-1 14:05:01 16-Jul-86 Page 39
CLIP12M4A.FTN2 /F77/MR

SUBROUTINE TIES(JSUB,MM,PTIE)

0001 C COMPUTE PROBABILITY OF CORRECT DECISION GIVEN THAT
C SEVERAL SATURATED CHANNELS ARE TIED
C
0002 C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      P2LM(2:3),
0003      LOGICAL1 GO
      COMMON /DENPAR/ BICK, AAB, BAB, LJAM,
      $ COMMON /PARDEN/ TAU, TAU2, TAUK, TAUK2
      $ COMMON /QUES/ QD, QL
      C NUMBER OF NON-SIGNAL CHANNELS
      MMU=MM-1
0004 PTIE=0.0 DC
0005 CUE0=DEXP(-TAU)
0006 CUE1=DEXP(-TAUK)
0007 P1L=DXI((QD-2)*JSUB(1))*DXI(Q1,JSUB(1))
0008 DO 10 I=2,MM
0009 P2LM(I)=DXI(CUE0,2-JSUB(1))*DXI(CUE1,JSUB(1))
0010 CONTINUE
0011
0012
0013
0014
0015 C SET UP VECTOR LOOP PARAMETERS
0016 DO 20 I=1,MM-1
0017 LL0W(I)=0
0018 LINC(I)=1
0019 LUP(I)=1
0020 CONTINUE
0021 PTIE=0.0 DO
0022 CALL VLINIT(MU,LL0W,MM-1)
0023 MU=0
0024 MU=0
0025 MU=0
0026 CONTINUE
0027 FRAC=1.0/(1.00+MU)
0028 PROD=1.00
0029 DO 50 M=2,MM
0030 IF (MU(M-1).EQ.1) THEN
      PROD=PROD*P2LM(M)
0031 ELSE
      PROD=PROD*(1.00-P2LM(M))
0032 END IF
0033 CONTINUE
0034 PTIE=FRAC*PROD*PTIE
0035 CALL VLITER(MU,LL0W,LUP,LINC,MM-1,60)
0036 IF (GO) GOTO 30
0037 PTIE=PTIE*PT1L
0038 RETURN
0040 END

```

```

PDF-11 FORTRAN-77 V4.0-1 14:05:06 16-Jul-86 Page 40
CLIP12M4A.FTN2 /F77/MR

0001 C FUNCTION FOR UNJAMMED PTIE FOR OPT. THRESHOLD SEARCH
C NOTE: WHEN JAMMING EVENT IS (0,0,...,0), THE VARIABLES
C RICK, AAB, BAB, TAUK, AND TAUK2 ARE NOT
C USED IN THE COMPUTATIONS.
C
0002 C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION NOJAM(4)
      COMMON /INPUTS/ DEBRON(3),MSLOTS,K,MM
      COMMON /DENPAR/ BICK, AAB, BAB, LJAM,
      $ COMMON /PARDEN/ TAU, TAU2, TAUK, TAUK2
      DATA NOJAM/0.0,0.0,0/
      LJAM=G
      TAU=ETA
      TAU2=TAU4ETA
      CALL PSEL(NOJAM,MM,P)
      PUNJAM=P
      RETURN
      END

PDF-11 FORTRAN-77 V4.0-1 14:05:07 16-Jul-86 Page 41
CLIP12M4A.FTN2 /F77/MR

0001 C SUBROUTINE SETTAU(MM,PE00)
C SEARCH FOR OPTIMUM THRESHOLD IN ABSENCE OF JAMMING
C
0002 C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      EXTERNAL PUNJAM
      COMMON /DENPAR/ BICK, AAB, BAB, LJAM,
      $ COMMON /PARDEN/ RMM, RHOT
      COMMON /QUES/ QD, QL
      L2M=0
      C GUESS BASED ON QUADRATIC CURVE FIT
      IF (MM.EQ.2) THEN
      GUESS=0.92500*4-8.47500*2+32.45D0
      ELSE IF (MM.EQ.4) THEN
      GUESS=1.0500*4-9.3500*2+34.5D0
      ELSE IF (MM.EQ.8) THEN
      GUESS=1.100*4-9.900*2+36.3D0
      ELSE
      GUESS=15.0D0
      END IF
      CALL MINSE(PUNJAM,PE00,TAUOPT,1.00,GUESS,0.00,
      $ TAU=TAUOPT
      $ TAU2=TAU+TAU
      $ PE00=PEMIN
      RETURN
      END


```

```

0001      SUBROUTINE MINSER(F,FMIN,XMIN,STEP,GUESS,BLIM,ULIM,TOL)
C SEARCH FOR MINIMUM OF F(X) OVER THE INTERVAL BLIM <= X <= ULIM
C TROUBLE MAY OCCUR IF F(X) HAS MULTIPLE LOCAL MINIMA WITHIN THE
C SEARCH INTERVAL OR IF THE FUNCTION IS VERY STEEP AND STEP IS
C TOO BIG.
C
C   F = NAME OF FUNCTION TO BE MINIMIZED
C   FMIN = MINIMUM VALUE OF F(X) OVER INTERVAL
C   XMIN = VALUE OF X FOR WHICH FMIN OCCURS
C   STEP = INITIAL STEP SIZE FOR SEARCH
C   GUESS = INITIAL GUESS AT XMIN, BLIM <= GUESS <= ULIM
C   BLIM = LOWER LIMIT OF SEARCH INTERVAL
C   ULIM = UPPER LIMIT OF SEARCH INTERVAL
C   TOL = TOLERANCE ON XMIN; SEARCH STOPS WHEN DX < TOL
C
C NOTE: F MUST BE A DOUBLE PRECISION FUNCTION OF ONE DOUBLE PRECISION
C ARGUMENT. ANY PARAMETERS CAN BE PASSED FROM THE CALLER VIA
C A COMMON BLOCK.
C
C PROGRAMMER: ROBERT H. FRENCH DATE: 17 MARCH 1986
C
C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C   X=GUESS
C   SUMAX=DABS(X-BLIM)
C   SUMAX=DABS(ULIM-X)
C   DX=DMIN1(STEP,SUMAX)
C   TEST=TOL
C   0008      10   FO=F(X)
C   0009      11   F1=F(X+DX)
C
C ARE WE GOING IN THE RIGHT DIRECTION?
C   0010      12   IF(F1.LE.F0) GOTO 100
C   ...     NO, SWITCH DIRECTION
C   0011      13   DX=-DX
C   0012      14   F1=F(X+DX)
C   0013      15   IF(F1.LE.F0) GOTO 100
C
C ELSE WE MUST BE CLOSE TO A MIN. AT X=GUESS, SO CUT
C STEP SIZE AND TRY AGAIN
C   0014      16   DX=DX/10.00
C   0015      17   IF(DABS(DX).GE.TEST) GOTO 10
C
C CLOSE ENOUGH AT GUESS
C   0016      18   XMIN=X
C   0017      19   FMIN=F0
C
C NOW GOING RIGHT DIRECTION.
C KEEP GOING UNTIL PAST MINIMUM BY ONE STEP.
C
C   0019      20   X2=X+DX+DX
C   ...     HAVE WE REACHED END POINT?
C   0020      21   IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110

```

```

C ALL OK
C   0021      22   105   F2=F1(X2)
C   PAST MIN?
C   0022      23   IF(F2.GE.F1) GOTO 200
C   ... NO, STEP AGAIN
C   0023      24   FO=F1
C   0024      25   F1=F2
C   X=X+DX
C   0025      26   GOTO 100
C   MIN MAY BE AT AN ENDPOINT. CUT STEP SIZE AND TRY AGAIN
C   C IF INCREMENT NOT TOO SMALL.
C   110   IF(DABS(DX).LE.TEST) GOTO 120
C   X=X+DX
C   0027      27   FO=F1
C   0028      28   F1=F(X+DX)
C   0029      29   DX=DX/10.00
C   0030      30   F1=F(X+DX)
C   0031      31   GOTO 100
C   MIN MUST BE AT THE ENDPOINT (OR WITHIN MINIMUM DX THEREOF)
C   0032      32   120   IF(X2.LE.BLIM) GOTO 122
C   C MIN AT X=ULIM
C   0033      33   122   IF(X2.LE.BLIM) GOTO 122
C   C MIN AT X=ULIM
C   0034      34   XMIN=ULIM
C   0035      35   121   FMIN=F(XMIN)
C   RETURN
C   C MIN AT BLIM
C   0036      36   122   XMIN=BLIM
C   0037      37   GOTO 121
C   0038      38   XMIN=BLIM
C   HAVE PASSED MIN. IS IT LOCATED CLOSELY ENOUGH YET?
C   0039      39   200   IF(DABS(DX).LE.TEST) GOTO 300
C   ... NO, CUT STEP SIZE AND TRY AGAIN
C   0040      40   GOTO 115
C   DONE!
C   SINCE FO >= F1 & F2>= F1 AND ABS(DX)>MIN. DX, CALL F1 THE MIN.
C   0041      41   300   FMIN=F1
C   0042      42   XMIN=X+DX
C   RETURN
C   END
C
C   0043      43   300
C   0044      44   300
C
C   0045      45   300
C
C   0046      46   300
C
C   0047      47   300
C
C   0048      48   300
C
C   0049      49   300
C
C   0050      50   300
C
C   0051      51   300
C
C   0052      52   300
C
C   0053      53   300
C
C   0054      54   300
C
C   0055      55   300
C
C   0056      56   300
C
C   0057      57   300
C
C   0058      58   300
C
C   0059      59   300
C
C   0060      60   300
C
C   0061      61   300
C
C   0062      62   300
C
C   0063      63   300
C
C   0064      64   300
C
C   0065      65   300
C
C   0066      66   300
C
C   0067      67   300
C
C   0068      68   300
C
C   0069      69   300
C
C   0070      70   300
C
C   0071      71   300
C
C   0072      72   300
C
C   0073      73   300
C
C   0074      74   300
C
C   0075      75   300
C
C   0076      76   300
C
C   0077      77   300
C
C   0078      78   300
C
C   0079      79   300
C
C   0080      80   300
C
C   0081      81   300
C
C   0082      82   300
C
C   0083      83   300
C
C   0084      84   300
C
C   0085      85   300
C
C   0086      86   300
C
C   0087      87   300
C
C   0088      88   300
C
C   0089      89   300
C
C   0090      90   300
C
C   0091      91   300
C
C   0092      92   300
C
C   0093      93   300
C
C   0094      94   300
C
C   0095      95   300
C
C   0096      96   300
C
C   0097      97   300
C
C   0098      98   300
C
C   0099      99   300
C
C   0100      100   300
C
C   0101      101   300
C
C   0102      102   300
C
C   0103      103   300
C
C   0104      104   300
C
C   0105      105   300
C
C   0106      106   300
C
C   0107      107   300
C
C   0108      108   300
C
C   0109      109   300
C
C   0110      110   300
C
C   0111      111   300
C
C   0112      112   300
C
C   0113      113   300
C
C   0114      114   300
C
C   0115      115   300
C
C   0116      116   300
C
C   0117      117   300
C
C   0118      118   300
C
C   0119      119   300
C
C   0120      120   300
C
C   0121      121   300
C
C   0122      122   300
C
C   0123      123   300
C
C   0124      124   300
C
C   0125      125   300
C
C   0126      126   300
C
C   0127      127   300
C
C   0128      128   300
C
C   0129      129   300
C
C   0130      130   300
C
C   0131      131   300
C
C   0132      132   300
C
C   0133      133   300
C
C   0134      134   300
C
C   0135      135   300
C
C   0136      136   300
C
C   0137      137   300
C
C   0138      138   300
C
C   0139      139   300
C
C   0140      140   300
C
C   0141      141   300
C
C   0142      142   300
C
C   0143      143   300
C
C   0144      144   300
C
C   0145      145   300
C
C   0146      146   300
C
C   0147      147   300
C
C   0148      148   300
C
C   0149      149   300
C
C   0150      150   300
C
C   0151      151   300
C
C   0152      152   300
C
C   0153      153   300
C
C   0154      154   300
C
C   0155      155   300
C
C   0156      156   300
C
C   0157      157   300
C
C   0158      158   300
C
C   0159      159   300
C
C   0160      160   300
C
C   0161      161   300
C
C   0162      162   300
C
C   0163      163   300
C
C   0164      164   300
C
C   0165      165   300
C
C   0166      166   300
C
C   0167      167   300
C
C   0168      168   300
C
C   0169      169   300
C
C   0170      170   300
C
C   0171      171   300
C
C   0172      172   300
C
C   0173      173   300
C
C   0174      174   300
C
C   0175      175   300
C
C   0176      176   300
C
C   0177      177   300
C
C   0178      178   300
C
C   0179      179   300
C
C   0180      180   300
C
C   0181      181   300
C
C   0182      182   300
C
C   0183      183   300
C
C   0184      184   300
C
C   0185      185   300
C
C   0186      186   300
C
C   0187      187   300
C
C   0188      188   300
C
C   0189      189   300
C
C   0190      190   300
C
C   0191      191   300
C
C   0192      192   300
C
C   0193      193   300
C
C   0194      194   300
C
C   0195      195   300
C
C   0196      196   300
C
C   0197      197   300
C
C   0198      198   300
C
C   0199      199   300
C
C   0200      200   300
C
C   0201      201   300
C
C   0202      202   300
C
C   0203      203   300
C
C   0204      204   300
C
C   0205      205   300
C
C   0206      206   300
C
C   0207      207   300
C
C   0208      208   300
C
C   0209      209   300
C
C   0210      210   300
C
C   0211      211   300
C
C   0212      212   300
C
C   0213      213   300
C
C   0214      214   300
C
C   0215      215   300
C
C   0216      216   300
C
C   0217      217   300
C
C   0218      218   300
C
C   0219      219   300
C
C   0220      220   300
C
C   0221      221   300
C
C   0222      222   300
C
C   0223      223   300
C
C   0224      224   300
C
C   0225      225   300
C
C   0226      226   300
C
C   0227      227   300
C
C   0228      228   300
C
C   0229      229   300
C
C   0230      230   300
C
C   0231      231   300
C
C   0232      232   300
C
C   0233      233   300
C
C   0234      234   300
C
C   0235      235   300
C
C   0236      236   300
C
C   0237      237   300
C
C   0238      238   300
C
C   0239      239   300
C
C   0240      240   300
C
C   0241      241   300
C
C   0242      242   300
C
C   0243      243   300
C
C   0244      244   300
C
C   0245      245   300
C
C   0246      246   300
C
C   0247      247   300
C
C   0248      248   300
C
C   0249      249   300
C
C   0250      250   300
C
C   0251      251   300
C
C   0252      252   300
C
C   0253      253   300
C
C   0254      254   300
C
C   0255      255   300
C
C   0256      256   300
C
C   0257      257   300
C
C   0258      258   300
C
C   0259      259   300
C
C   0260      260   300
C
C   0261      261   300
C
C   0262      262   300
C
C   0263      263   300
C
C   0264      264   300
C
C   0265      265   300
C
C   0266      266   300
C
C   0267      267   300
C
C   0268      268   300
C
C   0269      269   300
C
C   0270      270   300
C
C   0271      271   300
C
C   0272      272   300
C
C   0273      273   300
C
C   0274      274   300
C
C   0275      275   300
C
C   0276      276   300
C
C   0277      277   300
C
C   0278      278   300
C
C   0279      279   300
C
C   0280      280   300
C
C   0281      281   300
C
C   0282      282   300
C
C   0283      283   300
C
C   0284      284   300
C
C   0285      285   300
C
C   0286      286   300
C
C   0287      287   300
C
C   0288      288   300
C
C   0289      289   300
C
C   0290      290   300
C
C   0291      291   300
C
C   0292      292   300
C
C   0293      293   300
C
C   0294      294   300
C
C   0295      295   300
C
C   0296      296   300
C
C   0297      297   300
C
C   0298      298   300
C
C   0299      299   300
C
C   0300      300   300
C
C   0301      301   300
C
C   0302      302   300
C
C   0303      303   300
C
C   0304      304   300
C
C   0305      305   300
C
C   0306      306   300
C
C   0307      307   300
C
C   0308      308   300
C
C   0309      309   300
C
C   0310      310   300
C
C   0311      311   300
C
C   0312      312   300
C
C   0313      313   300
C
C   0314      314   300
C
C   0315      315   300
C
C   0316      316   300
C
C   0317      317   300
C
C   0318      318   300
C
C   0319      319   300
C
C   0320      320   300
C
C   0321      321   300
C
C   0322      322   300
C
C   0323      323   300
C
C   0324      324   300
C
C   0325      325   300
C
C   0326      326   300
C
C   0327      327   300
C
C   0328      328   300
C
C   0329      329   300
C
C   0330      330   300
C
C   0331      331   300
C
C   0332      332   300
C
C   0333      333   300
C
C   0334      334   300
C
C   0335      335   300
C
C   0336      336   300
C
C   0337      337   300
C
C   0338      338   300
C
C   0339      339   300
C
C   0340      340   300
C
C   0341      341   300
C
C   0342      342   300
C
C   0343      343   300
C
C   0344      344   300
C
C   0345      345   300
C
C   0346      346   300
C
C   0347      347   300
C
C   0348      348   300
C
C   0349      349   300
C
C   0350      350   300
C
C   0351      351   300
C
C   0352      352   300
C
C   0353      353   300
C
C   0354      354   300
C
C   0355      355   300
C
C   0356      356   300
C
C   0357      357   300
C
C   0358      358   300
C
C   0359      359   300
C
C   0360      360   300
C
C   0361      361   300
C
C   0362      362   300
C
C   0363      363   300
C
C   0364      364   300
C
C   0365      365   300
C
C   0366      366   300
C
C   0367      367   300
C
C   0368      368   300
C
C   0369      369   300
C
C   0370      370   300
C
C   0371      371   300
C
C   0372      372   300
C
C   0373      373   300
C
C   0374      374   300
C
C   0375      375   300
C
C   0376      376   300
C
C   0377      377   300
C
C   0378      378   300
C
C   0379      379   300
C
C   0380      380   300
C
C   0381      381   300
C
C   0382      382   300
C
C   0383      383   300
C
C   0384      384   300
C
C   0385      385   300
C
C   0386      386   300
C
C   0387      387   300
C
C   0388      388   300
C
C   0389      389   300
C
C   0390      390   300
C
C   0391      391   300
C
C   0392      392   300
C
C   0393      393   300
C
C   0394      394   300
C
C   0395      395   300
C
C   0396      396   300
C
C   0397      397   300
C
C   0398      398   300
C
C   0399      399   300
C
C   0400      400   300
C
C   0401      401   300
C
C   0402      402   300
C
C   0403      403   300
C
C   0404      404   300
C
C   0405      405   300
C
C   0406      406   300
C
C   0407      407   300
C
C   0408      408   300
C
C   0409      409   300
C
C   0410      410   300
C
C   0411      411   300
C
C   0412      412   300
C
C   0413      413   300
C
C   0414      414   300
C
C   0415      415   300
C
C   0416      416   300
C
C   0417      417   300
C
C   0418      418   300
C
C   0419      419   300
C
C   0420      420   300
C
C   0421      421   300
C
C   0422      422   300
C
C   0423      423   300
C
C   0424      424   300
C
C   0425      425   300
C
C   0426      426   300
C
C   0427      427   300
C
C   0428      428   300
C
C   0429      429   300
C
C   0430      430   300
C
C   0431      431   300
C
C   0432      432   300
C
C   0433      433   300
C
C   0434      434   300
C
C   0435      435   300
C
C   0436      436   300
C
C   0437      437   300
C
C   0438      438   300
C
C   0439      439   300
C
C   0440      440   300
C
C   0441      441   300
C
C   0442      442   300
C
C   0443      443   300
C
C   0444      444   300
C
C   0445      445   300
C
C   0446      446   300
C
C   0447      447   300
C
C   0448      448   300
C
C   0449      449   300
C
C   0450      450   300
C
C   0451      451   300
C
C   0452      452   300
C
C   0453      453   300
C
C   0454      454   300
C
C   0455      455   300
C
C   0456      456   300
C
C   0457      457   300
C
C   0458      458   300
C
C   0459      459   300
C
C   0460      460   300
C
C   0461      461   300
C
C   0462      462   300
C
C   0463      463   300
C
C   0464      464   300
C
C   0465      465   300
C
C   0466      466   300
C
C   0467      467   300
C
C   0468      468   300
C
C   0469      469   300
C
C   0470      470   300
C
C   0471      471   300
C
C   0472      472   300
C
C   0473      473   300
C
C   0474      474   300
C
C   0475      475   300
C
C   0476      476   300
C
C   0477      477   300
C
C   0478      478   300
C
C   0479      479   300
C
C   0480      480   300
C
C   0481      481   300
C
C   0482      482   300
C
C   0483      483   300
C
C   0484      484   300
C
C   0485      485   300
C
C   0486      486   300
C
C   0487      487   300
C
C   0488      488   300
C
C   0489      489   300
C
C   0490      490   300
C
C   0491      491   300
C
C   0492      492   300
C
C   0493      493   300
C
C   0494      494   300
C
C   0495      495   300
C
C   0496      496   300
C
C   0497      497   300
C
C   0498      498   300
C
C   0499      499   300
C
C   0500      500   300
C
C   0501      501   300
C
C   0502      502   300
C
C   0503      503   300
C
C   0504      504   300
C
C   0505      505   300
C
C   0506      506   300
C
C   0507      507   300
C
C   0508      508   300
C
C   0509      509   300
C
C   0510      510   300
C
C   0511      511   300
C
C   0512      512   300
C
C   0513      513   300
C
C   0514      514   300
C
C   0515      515   300
C
C   0516      516   300
C
C   0517      517   300
C
C   0518      518   300
C
C   0519      519   300
C
C   0520      520   300
C
C   0521      521   300
C
C   0522      522   300
C
C   0523      523   300
C
C   0524      524   300
C
C   0525      525   300
C
C   0526      526   300
C
C   0527      527   300
C
C   0528      528   300
C
C   0529      529   300
C
C   0530      530   300
C
C   0531      531   300
C
C   0532      532   300
C
C   0533      533   300
C
C   0534      534   300
C
C   
```


J. S. LEE ASSOCIATES, INC.

APPENDIX J COMPUTER PROGRAM FOR CLIPPER RECEIVER WITH L=3 HOPS/SYMBOL

The following pages contain the source code listing for the FORTRAN-77 program used for numerical computations of the performance of the clipper receiver when L=3 hops/symbol, using a numerical search for the worst-case jamming fractions. If M>4 the sizes of the arrays used in computing event probabilities must be increased and the corresponding array-size parameters in calls to PUTIN and LOOKUP changed accordingly.

```

0001      PROGRAM CLIP3R
C THIS PROGRAM COMPUTES THE ERROR PROBABILITY FOR RANDOM M-ARY
C FSK/FH WITH 3 HOPS/BIT IN THE PRESENCE OF PARTIAL-BAND
C NOISE JAMMING BY NUMERICAL INTEGRATION FOR THE CLIPPER RECEIVER
C AND BETWEEN-CONDITIONAL-PROB. RESTART CAPABILITY
C ANALYSIS: L. E. MILLER, R. H. FRENCH
C PROGRAM: R. H. FRENCH
C V 3.1.0 - COMPUTATIONS ONLY
C
0002      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003      PARAMETER (LJ=11)
0004      CHARACTER*13 FNAME, GRNAME
0005      LOGICAL DOTAU, TEST
0006      LOGICAL*16 GOOD, RESTRT
0007      DIMENSION POFQ(50), ION(50)
0008      REAL*4 PRLOG(LJ), DBSR(LJ), OPT(LJ)
0009      VIRTUAL A(100), ISUB(100), C(625), ISUB(625)
0010      VIRTUAL D(625), ISUB(625), PRERR(625), PSUB(625)
0011      C COMMON /INPUTS/ PASSES PARAMETERS OF THE RUN
0012      COMMON /INPUTS/ DEBNOL(3) , NSLOTS,K,MM
0013      COMMON /SIZE/ PASSES NUMBERS OF PARAMETERS
0014      COMMON /DENPAR/ BIGK, AAB, BAB, LJAM, AAB2, BAD2, ABBAB,
     J-2          TAU, TAU2, TAU3, TAUK, TAUK2, TAU3
0015      COMMON /PARDEN/ RHOM, RMOT
0016      COMMON /RSPAR/ JAM(4), JAM1, MPS, LJ, RESTRT
0017      DATA YES, NO, BLANK /'Y', 'N', ' ', '/'/
0018      CALL JS60
0019      CALL ERSET29, TRUE, FALSE, .TRUE., .FALSE., .15)
0020      CALL GET(LJ, START, DBINC)
0021      SLOTS=NSLOTS
0022      NORBIT=0.500**MM/(MM-1.00)
0023      DO 800 LJ=1,NO
0024      DOTAU=.TRUE.
0025      EBNO=10.00***(DEBNOL(10)/10.00)
0026      RMOM=K*EBNO/3.00
0027      IOOUT=DEBNOL(10)

C OPEN DATA FILE
C
0028      WRITE(FNAME,730) MM, IOOUT
0029      FORMAT(100J11,13I12.2,'.DAT')
0030      WRITE(6,776) MM, DEBNOL(10)
0031      776 FORMAT(1CLIPPER RECEIVER, OPTIMUM GAMMA RESULTS'/'
     ,M=12.5X,L=315X,'EB/NO='FR.A/,'EB/N0' (dB)',/
     ,5X,'P(e)',15X,'Opt.')
0032      WRITE(5,731) FNAME
0033      733 FORMAT(1WORKING ON 'A13)
0034      OPEN(UNIT=6,FILE=FILENAME,STATUS='OLD',FORM='UNFORMATTED',
     S, ERR=750)

```

POP-111 FORTRAN-77 V4.0-1 10:43:57 16-Jul-86
 CLIP3SGR.FTN:6 /F77/MR

```

0067 Q=0OPT(I,J-1)          10:43:57 16-Jul-86
0068   IQ=Q                  Page 3
0069   ELSE
0070     100=1
0071     Q=1.00
0072     IQ=1
0073   END IF
0074   IQ=100
0075   C GIVE PROGRESS MESSAGE TO T:
0076   WRITE(5,601) IJ
0076   FORMAT(1I3,1I3)
0076   DEBNJ=START+(IJ-1)*DEBNJ
0076   DEBSJR(IJ)=DEBNJ
0077   R=10.0D+0*DEBNJ/10.0D
0078   DO 602 105=1,50
0079     POFQ(IQS)=0.0D
0080   CONTINUE
0081   IQS=0
0082   IQS=0
0083   602
0084   C PRIME THE ALGORITHM WITH DUMMY OLD VALUES OF P(E)
0085   P1=0.0D
0086   P2=0.0D
0087   709  GAMMA/Q/SLOTS
0088   WRITE(GNAME,735) MN,10
0088   FORMAT('EQ',11,'3','14.4','DAT')
0089   OPEN(UNIT=2,FILE=GNAME,STATUS='OLD',FORM='UNFORMATTED',
0090     READING,ERR=770)
0091   WRITE(5,*3939;
0092   FORMAT(' READING EVENT FILE')
0093   READ(3,D10SUB,MUSED,GOOD)
0094   CLOSE(UNIT=3)
0095   C IF FILE FOR EVENT PROBABILITIES DOES NOT EXIST, CALCULATE THEM
0095   C AND CREATE A FILE.
0096   770  CONTINUE
0097   WRITE(5,*3938)
0098   FORMAT(' CREATING EVENT FILE')
0098   CALL GENPTE(MN,IQ,MSLOTS,GOOD,A,TASUB,
0098   C,ICSUB,D,IDSUB,MUSED)
0099   S  OPEN(UNIT=3,FILE=GNAME,STATUS='NEW',FORM='UNFORMATTED')
0100   WRITE(3,D10SUB,MUSED,GOOD)
0101   CLOSE(UNIT=3)
0102   IF( NOT(SGOOD)) GOTO 700
0103   RHOTSGAMMA=R*EBNO/(GAMMA+R*EBNO)
0104   RHOT=R*K*RHOITS/3.0D
0105   C EVALUATE THE PROBABILITY
0106   DO 780 MDS=1,IQS
0107   IF( TOL(MDS).EQ.10) THEN
0108     PESTM=POFQ(MDS)
0109     GOTO 781
0110   END IF
0111   CONTINUE
  
```

POP-111 FORTRAN-77 V4.0-1 10:43:57 16-Jul-86
 CLIP3SGR.FTN:6 /F77/MR

```

C NOT IN STORED LIST, COMPUTE IT
0112   CALL PSUBE(MN,PESTM,D,IDSUB,MUSED,PRERR,IPSUB,PECO)
0113   TOS=105+1
0114   IF(105.GT.50) THEN
0115     TOS=50
0116   ELSE
0117     POFQ(IQS)=PESYM
0118     TOL(IQS)=IQ
0119   END IF
0120   781  P3=PESYM
0121     IF(IP3.GT.P2 .AND. 10.ILT.MSLOTS) THEN
0122       C KEEP ON GOING, WE ARE NOT PAST THE MAXIMUM
0123     P1=P2
0124     P2=P3
0125     IQ=MIN((IQ+10Q,MSLOTS))
0126     Q=DMIN((Q+9Q,MSLOTS))
0127   GOTO 709
0128   PMAX=DMAX1(P1,P2,P3)
0129   EPS=0.001D0*PMAX
0130   TEST=(DABS(P1-P2).LE.EPS) .AND. DABS(P1-P3).LE.EPS .AND.
0130   DABS(P2-P3).LE.EPS
0131   S  S
0131   IF( TEST .OR. IQ.EQ.1
0131   OR. ((.NOT.TEST) .AND. 10.EQ.MSLOTS)) THEN
0132     C  C
0132     WE ARE DONE WHEN ALL 3 ARE CLOSE TOGETHER OR WHEN DQ=1
0132     OR WHEN WE REACHED FULL-BAND JAMMING AND P(E) IS STILL
0132     INCREASING
0132     POP=PMAX
0132     IF(P2.GT.P3) THEN
0132       C  C
0132       THE OPTIMUM MUST BE THE MIDDLE POINT OF THE 3
0132       QOPT(IJ)=Q-DQ
0132       IF(QOPT(IJ).EQ.0.D0) QOPT(IJ)=1.D0
0133     C  C
0133     C PREVENT ROUND-OFF FROM MAKING QOPT VS. EB/NJ NON-MONOTOMIC
0133     IF(IJ.GT.1) THEN
0133       IF(QOPT(IJ).LT.QOPT(IJ-1)) QOPT(IJ)=QOPT(IJ-1)
0134   GOTO 665
0134   END IF
0135   ELSE
0135   THE OPTIMUM IS FULL-BAND JAMMING
0135   QOPT(IJ)=MSLOTS
0136   END IF
0136   C  C
0136   IF(QOPT(IJ).LT.QOPT(IJ-1)) QOPT(IJ)=QOPT(IJ-1)
0137   END IF
0137   C  C
0137   IF(QOPT(IJ).EQ.0.D0) QOPT(IJ)=1.D0
0138   C  C
0138   IF(QOPT(IJ).LT.QOPT(IJ-1)) QOPT(IJ)=QOPT(IJ-1)
0139   C  C
0139   C NOT LOCATED SUFFICIENTLY ACCURATELY, CUT DQ AND TRY AGAIN
0140   C  C
0140   C
0141   C
0142   C
0143   C
0144   C
0144   C
0145   C
0145   C
0146   C
0146   C
0147   C
0147   C
0148   C
0148   C
0149   C
0149   C
0150   C
0150   C
0151   C
0151   C
  
```

```

PDP-11 FORTRAN-77 V4.0-1    10:43:57   16-Jul-86   Page 5
CLIP13SGR.FTN:6      /F77/MR

0152      GOTO 709
0153      END IF
0154      END IF
0155      PE=MORBIT*POPT
0156      WRITE(6,666) DBSAR(IJ),PE,QQPT(IJ)
0157      FORMAT(1X,F7.3,S1.1P012.5,X,1PD12.5)
0158      PRLOG(IJ)=DLG610(PE)
0159      OPEN(UNIT=4,FILE=FFNAME,STATUS='OLD',ACCESS='APPEND',
0160           FORM='UNFORMATTED')
0161      WRITE(4, DBSAR(IJ), PRLOG(IJ), QQPT(IJ))
0162      CLOSE(UNIT=4)
0163      COMTIME
0164      OPEN(UNIT=4,FILE=FFNAME,STATUS='NEW',FORM='UNFORMATTED')
0165      WRITE(4, MM,3,DEBNOL(10),MSLOTS,DBSSR,PRLOG,QQPT
0166      CLOSE(UNIT=4)
0167      WRITE(6,776) MM,DEBNOL(10)
0168      DO 689 IJ=1,MJ
0169      WRITE(6,666) DBSAR(IJ),10.**PRLOG(IJ),QQPT(IJ)
0170      CONTINUE
0171      WRITE(6,688) TAU
0172      FORMAT(1///, OPTIMUM THRESHOLD FOR ABOVE IS ETA/SIGMA**2 =',
0173      800  CONTINUE
0174      900  CONTINUE
0175      STOP 0
0176      END
0177      700  CONTINUE
0178      -4
0179      800  CONTINUE
0180      900  CONTINUE
0181      STOP 0
0182      END

```

```

PDP-11 FORTRAN-77 V4.0-1    10:44:25   16-Jul-86   Page 6
CLIP13SGR.FTN:6      /F77/MR

0001      SUBROUTINE GETMJ,START,DBINC
0002      C  INTERACTIVE INPUT OF PARAMETERS FOR RUM
0003      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0004      CHARACTER FIELD,BLANK9
0005      COMMON /INPUTS/ DEBNOL(3),MSLOTS,K,MN
0006      COMMON /SIZE/ NO
0007      C DEFAULT LISTS TEMPORARILY NEEDED ARE IN SHARED STORAGE WITH
0008      C THE LARGE CONVOLUTION WORKING ARRAYS
0009      DATA BLANK9/          '/
0010      WRITE(5,33)
0011      FORMAT(1X,BITS/SYMBOL (K) [2]: '$')
0012      READ(5,31)
0013      1   FORMAT(1X,EQ.0)K=2
0014      2   FORMAT(1X, HOW MANY EB/MO? [1]: '$')
0015      READ(5,31)NO
0016      3   FORMAT(12)
0017      IF(MO.EQ.0)NO=1
0018      DO 7 1M=1,MO
0019      DO=DBNR(1M,K)
0020      4   WRITE(5,5)IN,DO
0021      5   FORMAT(1X,EB/MO(' J2.'): ',F9.6,']: '$')
0022      READ(5,6)FIELD
0023      FORMAT(A9)
0024      IF(FIELD.EQ.BLANK9) THEN
0025      DEBNOL(1M)=DO
0026      ELSE
0027      DECODE(9,6i,FIELD);DEBNOL(1M)
0028      FORMAT(F9.6)
0029      END IF
0030      CONTINUE
0031      MSLOTS=2400
0032      WRITE(5,39)
0033      FORMAT(1X, HOW MANY EB/MJ? [11]: '$')
0034      READ(5,34)MJ
0035      FORMAT(13)
0036      IF(MJ.EQ.0) MJ=11
0037      IF(MJ.LT.0.OR. MJ.GT.11) 6070 32
0038      WRITE(5,41)
0039      FORMAT(1X, STARTING VALUE FOR EB/MJ (DB) [50.]: '$')
0040      READ(5,42,ERR=40) START
0041      IF(START.EQ.0.00) START=50.00
0042      WRITE(5,36)
0043      IF(MJ.EQ.1) RETURN
0044      35   FORMAT(1X,DB INCREMENT FOR EB/MJ [-5.0]: '$')
0045      READ(5,37,ERR=35) DBINC
0046      37   FORMAT(F6.3)
0047      IF(DBINC.EQ.0.00) DBINC=-5.00
0048      RETURN
0049      END
0050

```

```

0001      SUBROUTINE PSUB(M,P,D,IOSUB,NUSED,PRERR,IPSUB,PEOO)
C COMPUTE UNCONDITIONAL ERROR PROBABILITY
C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      INTEGER LUP(4),JSUB(4)
      LOGICAL JGO,MORE,STORE,RESTRT
      C WE DO WANT TO STORE ZERO ELEMENTS OF THE DENSITY FUNCTION.
      C SINCE IT TAKES TIME TO AVOID REPEATING THE UNDERFLOWS.
      VIRTUAL PRERR(E25),IPSUB(625)
      VIRTUAL D(625),IOSUB(625)
      COMMON /RSPAR/ JAM(4),JAP1,MPS,1J,RESTRT
      COMMON /SHAREZ/ LM(4),LINC(4)
      COMMON /DEMPAR/ BISK,AAB,BAB,LJAM,AAB2,BAB2,AABBAB,
      TAU,TAU2,TAU3,TAUK,TANK2,TANK3
      COMMON /PARDEN/ RHON,RHOT
      DATA STORE/.TRUE./
      PE=0,DO
      0012   10 1=1,M
      0013   LUP(1)=3
      0014   JSUB(1)=0
      0015   CONTINUE
      0016   IF(.NOT.RESTRT) THEN
      0017     NP5=0
      0018   C THE ALL-ZERO JAMMING EVENT P(E) IS AVAILABLE FROM THE
      C SEARCH FOR THE OPTION THRESHOLD, SO PUT IT INTO THE
      C ARRAY OF SAVED VALUES.
      CALL LOCNM(LM,LUP,JSUB)
      CALL PUTIN(PEDO,PRERR,IPSUB,MPS,625,ISUB,KODE,STORE)
      IF(KODE.NE.0) STOP 'PRERR FULL.'
      JAN1=-1
      ELSE
        RESTRT BYPASSES VECTOR LOOP INITIALIZATION CALL
        GOTO 100
      END IF
      0025   C START VECTOR-INDEXED LOOP ON JAMMING EVENTS
      CALL VLINIT(JAM,LM,M)
      0026   CONTINUE
      0027   100   IF(JAM1.NE.JAM(1)) THEN
      0028     C UPDATE TEST VALUE FOR NEXT TIME, AND ...
      0029     JAN1=JAM(1)
      END IF
      0030   CALL EVENT(M,JAM,PIE,D,IOSUB,NUSED)
      0031   C BYPASS CONDITIONAL ERROR PROBABILITY CALCULATION IF EVENT
      C PROBABILITY IS ZERO. THIS SAVES MUCH TIME.
      IF(PIE.EQ.0.0D0) GOTO 101
      0032   C SINCE JAMMING PROBABILITIES DEPEND ONLY ON NO. OF CHANNELS
      C HAVING JAM(1) HOPS JAMMED AND NOT THE ARRANGEMENT OF THE
      C CHANNELS, WE CAN SORT THE NON-SIGNAL CHANNELS INTO ASCENDING
      C NUMBERS OF HOPS JAMMED. THIS REDUCES NUMBER OF DISTINCT
      C CONDITIONAL ERROR PROBABILITIES WHICH MUST BE SAVED TO AVOID
      C RECOMPUTING THEM UNNECESSARILY.

```

```

0033   DO 111 1=1,M
      JSUB(1)=JAM(1)
0034   CONTINUE
      IF(M.EQ.2) GOTO 199
0035   111
      0036   0037   DO 110 1=2,M-1
      0038   0039   IF((JSUB(1).LT.JSUB(1))) THEN
      0040     JTEMP=JSUB(1)
      0041     JSUB(1)=JSUB(1)
      0042     JSUB(1)=JTEMP
      0043   END IF
      0044   120   CONTINUE
      0045   110   CONTINUE
      0046   199   CONTINUE
      C STORE AS ELEMENT OF A SPARSE ARRAY TO SAVE MEMORY
      C EVEN THOUGH WE STORE ZEROS. THE SORTING OF SUBSCRIPTS
      C CUTS OUT MANY ELEMENTS.
      CALL LOCNM(LM,LUP,JSUB)
      C TRY TO FIND CONDITIONAL ERROR PROBABILITY IN STORED ARRAY
      CALL LOOKUP(PRERR,IPSUB,MPS,625,ISUB,STORE,MONE)
      0047   C IF IT IS NOT THERE, WE MUST COMPUTE IT
      0048   IF(MONE) THEN
      0049     CALL PSEL(JSUB,M,PROB)
      0050   C ... AND SAVE IT FOR POSSIBLE FUTURE RE-USE
      0051     CALL PUTIN(PRERR,IPSUB,MPS,625,ISUB,KODE,STORE)
      0052   IF(KODE.NE.0) STOP 2
      0053   END IF
      C SUM UP UNCONDITIONAL ERROR PROBABILITY
      PE=PE+PIE*PROB
      0054   C SAVE WHAT WE HAVE SO FAR FOR RESTART CAPABILITY
      C OPEN(UNIT=4,FILE='RESUME.DAT',STATUS='UNKNOWN',
      0055   5 FORM='UNFORMATTED')
      WRITE(4) JAM,JAP1,PRERR,MPS,JSUB,1J
      CLOSE(UNIT=4)
      0056   C ONCE WE HAVE UPDATED RESTART INFORMATION, THE RESTART
      C FLAG BECOMES FALSE.
      RESTRT=.FALSE.
      C ITERATE THE VECTOR-INDEX LOOP
      0057   101   CALL VLITER(LM,LUP,LINC,M,G0)
      IF(G0) GOTO 100
      RETURN
      END
0058
0059
0060
0061
0062

```

```

0001      SUBROUTINE EVENT(M,JAM,PIE,D,IDSUB,MUSED)
          C SUBROUTINE TO LOOK UP EVENT PROBABILITY FROM STORED ARRAY
          C
          C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          LOGICAL*1 STORE,MONE
          DIMENSION JAM(4),LUP(4)
          VIRTUAL D1(625),IDSUB(625)
          COMMON /SHARE2/LOW(4),LINC(4)
          DATA STORE/.FALSE./
          C SET UP ARRAY DESCRIPTION D(0:L,...,0:LL) WITH M DIMENSIONS
          DO 1 I=1,M
          LUP(I)=3
  1 CONTINUE
          C COMPUTE LINEAR EQUIVALENT SUBSCRIPT FOR JAMMING EVENT
          CALL LOCNM_1,0,LUP,JAM,IDSUB
          C LOOK UP THE VALUE, GET 0, DO IF NOT THERE
          CALL LOOKUP(PIE,D,IDSUB,MUSED,625,1SUB,MONE)
          RETURN
  0013
  0014

```

```

          C SUBROUTINE TO GENERATE EVENT PROBABILITIES
          C
          C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
          LOGICAL*1 GO,EGO,STORE,MONE,M000
          DIMENSION LUP2(4),LUP3(4)
          DIMENSION TUPA(4)
          DIMENSION TUP0(4)
          DIMENSION LUP1(4)
          VIRTUAL A(100),JASUB(100),C(625),ICSUM(625),
          S   D(625),IDSUB(625)
  0002
  0003
  0004
  0005
  0006
  0007
  0008
  0009
  0010
  0011
  0012
  0013
  0014
  0015
  0016
  0017
  0018
  0019
  0020
  0021
  0022
  0023
  0024
  0025
  0026
  0027
  0028
  0029
  0030
  0031
  0032
  0033

          C SUBROUTINE GEMPIE(MQ,MSLOTS,GOOD,A,IASUB,C,ICSUB,
          D,IDSUB,MUSED)

          C
          C SUBROUTINE TO GENERATE EVENT PROBABILITIES
          C
          C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
          LOGICAL*1 GO,EGO,STORE,MONE,M000
          DIMENSION LUP2(4),LUP3(4)
          DIMENSION TUPA(4)
          DIMENSION TUP0(4)
          DIMENSION LUP1(4)
          VIRTUAL A(100),JASUB(100),C(625),ICSUM(625),
          S   D(625),IDSUB(625)
          C SHARED STORAGE FOR COMMONLY NEEDED CONSTANT ARRAYS
          COMMON /SHARE2/L0(4),LINC(4)
          C SHARED STORAGE FOR: (1) INPUT DEFAULT LISTS, (2) CONDITIONAL PROB GEN.,
          C AND (3) EVENT PROB. GEN. THESE ARE NON-OVERLAPPING USAGES,
          COMMON /SHARE/ LUP2,LUP3,JUPD,L1,MUSEA,MUSEC,TERR
          1SUB,ISUB1,ISUB2,AIN,1,I1,III,MN,AOUT,CIN,CONT,
          S   DOUT,DIN
          C
          DATA I100/100/
          DATA TUPA/4*1/
          DATA LUP1/4*1/
          C STORE=.FALSE. --> DON'T STORE ZERO ELEMENTS OF SPARSE ARRAY.
          STORE=.FALSE.
          C
          IF(MQ.LE.0) THEN
          GOOD=.TRUE.
          IF(MQ.LE.0) THEN
          GOOD=.FALSE.
          RETURN
          END IF
          DO 80 LI=1,M
          TUP0(LI)=3
  80 CONTINUE
          C JAMMING PATTERN W/MON-ZERO PROBABILITY ON PER-HOP BASIS
          MUSEA=0
          C INITIALIZE VECTOR-INDEX LOOP
          CALL VLINIT(1,1,M,M)
          CONTINUE
          CALL LOCN(MQ,LOW,TUPA,1,1SUB)
          CALL PRHOP(1,MQ,MSLOTS,AIN)
          CALL PUTTM(AIN,A,IASUB,MUSEA,1100,1SUB,TERR,STORE)
          IF(TERR.NE.0)STOP 3
          C ITERATE VECTOR-INDEX LOOP
          CALL VLITER(1,LOW,LUP1,LINC,MN,GO)
          IF(GO) GOTO 90
          C COMPUTATION STARTS HERE. FIRST COPY A INTO D.
          C SINCE ARRAYS ARE A(0:1:0,1:1:0,1) AND D(0:1:0,1:1:0,1)
          C THE COPYING MUST BE DONE ON BASIS OF EQUIVALENT LINEAR
          C SUBSCRIPTS RATHER THAN A SIMPLE MOVE OPERATION.
          MUSED=0

```

PDP-11 FORTRAN-77 V4.0-1 10:44:40 16-Jul-86
 CLIP3SGR.FTN:6 /F77/MR

Page 11

```

C INITIALIZE VECTOR-INDEX LOOP
CALL VLINIT(I,LOW,MM)
CONTINUE
CALL LOCN(LOW,LOW,IUPA,J,ISUB1)
CALL LOCN(LOW,LOW,IUPD,J,ISUB2)
CALL LOOKUP(AOUT,A,IASUB,MUSEA,1100,ISUB1,STORE,MONE)
CALL PUTIN(DOUT,D,IDSUB,MUSED,625,ISUB2,JERR,STORE)
CALL VLITER(I,LOW,LUP1,LINC,MM,60)
IF(GO) GOTO 99
C ... L-1 CONVERGENCE ARE NEEDED ...
0042     DO 9998 L1=1,2
0043     DO 125 MM=1,MM
0044     CALL VLINIT(I1,LOW,MM)
0045     125
0046     MUSE=0
0047     CALL VLINIT(I,LOW,MM)
0048     CONTINUE
0049     CALL VLINIT(I1,LOW,MM)
0050     CONTINUE
0051     CALL LOCN(LOW,LOW,IUPA,J,ISUB1)
0052     CALL LOCN(LOW,LOW,IUPD,J,ISUB2)
0053     DO 21 MM=1,MM
0054     I1(MM)=I(MM)+I1(MM)
0055     CONTINUE
0056     CALL LOCN(LOW,LOW,IUPD,I1,ISUB3)
0057     CALL LOOKUP(AOUT,A,IASUB,MUSEA,1100,ISUB1,STORE,MONE)
0058     CALL LOOKUP(DOUT,D,IDSUB,MUSED,625,ISUB2,STORE,MONE)
0059     CALL LOOKUP(COUT,C,ICSUB,MSEC,625,ISUB3,STORE,MONE)
0060     CIN=COUT-AOUT*DOUT
0061     CALL PUTIN(CIN,C,ICSUB,MUSEC,625,ISUB3,JERR,STORE)
0062     IF(IERR.NE.0) STOP 4
C ITERATE VECTOR-LOOP FOR ARRAY D
0063     CALL VLITER(I1,LOW,LUP2,LINC,MM,602)
0064     IF(GO2) GOTO 97
C ITERATE VECTOR-LOOP FOR ARRAY A
0065     CALL VLITER(I,LOW,LUP1,LINC,MM,60)
0066     IF(GO) GOTO 98
C COPY C TO D IN SORTED ORDER FOR NEXT ITERATION
0067     MUSED=0
0068     CALL VLINIT(I1,LOW,MM)
0069     CONTINUE
0070     CALL LOCN(LOW,LOW,IUPD,J1,ISUB)
0071     CALL LOOKUP(COUT,C,ICSUB,MUSEC,625,ISUB,STORE,MONE)
0072     DIN=DOUT
0073     CALL PUTIN(DIN,D,IDSUB,MUSED,625,ISUB,JERR,STORE)
0074     IF(IERR.NE.0) STOP 5
0075     CALL VLITER(I1,LOW,LUP3,LINC,MM,60)
0076     IF(GO) GOTO 96
0077     CONTINUE
0078     RETURN
0079     END

```

J-7

PDP-11 FORTRAN-77 V4.0-1 10:44:48 16-Jul-86
 CLIP3SGR.FTN:6 /F77/MR

Page 12

```

0001     SUBROUTINE PUTIN(CIN,C,ICSUB,MUSE,MMAX,K,JERR,STORE)
C THIS SUBROUTINE INSERTS AN ELEMENT INTO A SPARSE ARRAY FOR
C WHICH ONLY THE NON-ZERO ELEMENTS ARE KEPT IN STORAGE IF
C THE SWITCH STORE IS .TRUE.
C THE DOUBLE PRECISION VALUE CIN IS STORED AS C(K) WHERE
C THE AUXILIARY ARRAY ICSUB IS USED TO KEEP TRACK OF THE
C SUBSCRIPTS OF THE CORRESPONDING ENTRIES IN C.
C
C USAGE:
C   LOGICAL*1 STORE
C   DOUBLE PRECISION C,CIN
C   VIRTUAL (ICSUB(MMAX)).C(MMAX)
C   CALL PUTIN(CIN,C,ICSUB,MUSE,MMAX,K,JERR,STORE)
C
C WHERE:
C   CIN = VALUE OF ELEMENT TO STORE
C   C = ARRAY IN WHICH NON-ZERO VALUES ARE ACTUALLY STORED
C   ICSUB = AUXILIARY ARRAY FOR ACTUAL SUBSCRIPT VALUES
C   MUSE = NUMBER OF ELEMENTS OF C CURRENTLY OCCUPIED
C   MMAX = SIZE OF ARRAY C
C   JERR = ERROR RETURN CODE. 0 IF NO ERROR OR 1 IF THERE IS
C   NO ROOM AVAILABLE IN C
C   STORE = .TRUE. TO STORE ZEROS EXPLICITLY. ELSE .FALSE.
C NOTE: IF CIN=0 AND THE SUBSCRIPT K IS FOUND IN ICSUB, THEN
C THE ELEMENT IS DELETED BY SHIFTING FORWARD ALL
C FOLLOWING ELEMENTS OF THE ARRAY
C
C PROGRAMMER: ROBERT H. FRENCH DATE: 11 JANUARY 1984
C
C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0002     VIRTUAL (ICSUB(MMAX)).C(MMAX)
0003     LOGICAL*1 STORE
0004     JERR=0
0005     IF(STORE) GOTO 5
0006     IF(CIN.EQ.0.0D0) GOTO 30
0007     IF(MUSE.EQ.0) GOTO 20
0008     5      DO 10 I=1,MUSE
0009     IF((ICSUB(I)).NE.K) GOTO 10
0010     C(I)=CIN
0011     RETURN
0012     CONTINUE
0013     10     IF(MUSE.LT.MMAX) GOTO 20
0014     0015     JERR=1
0016     RETURN
0017     20     MUSE=MUSE+1
0018     ICSUB(MUSE)=K
0019     C(MUSE)=CIN
0020     RETURN
0021     30     DO 40 I=1,MUSE
0022     J=J
0023     IF(ICSUB(I).EQ.K) GOTO 50
0024     CONTINUE

```

PDP-11 FORTRAN-77 V4.0-1 10:44:48 16-Jul-85
CLIP13SGR.FTN:6 /F77/MR

Page 13

```
0025      C RETURN
          C REMOVE THE ZEROED ELEMENT AND BUMP COUNT OF ENTRIES USED
          C
          0026      00 60 I=1,MUSE-1
          0027      ICSUB(I)=ICSUB(I+1)
          0028      C(I)-C(I-1)
          0029      CONTINUE
          0030      MUSE=MUSE-1
          0031      RETURN
          0032      END
```

```
0001      C
          C THIS SUBROUTINE RETRIEVES AN ELEMENT OF A SPARSE ARRAY WHICH
          C HAS BEEN STORED COMPACTLY BY STORING ONLY NON-ZERO ELEMENTS.
          C
          C THE ARRAY IS DOUBLE PRECISION.
          C
          C USAGE:
          C   VIRTUAL ICSUB(NMAX), C(NMAX)
          C   LOGICAL*1 STORE, NONE
          C   DOUBLE PRECISION COUNT
          C
          C CALL LOOKUP(COUT, C, ICSUB, N, NMAX, K, STORE, NONE)
          C
          C WHERE
          C   COUT = VALUE OF C(K) (OUTPUT FROM SUBROUTINE)
          C   C = ARRAY USED TO STORE NON-ZERO ELEMENTS
          C   ICSUB = AUXILIARY ARRAY TO STORE ACTUAL SUBSCRIPTS
          C   N = NUMBER OF ELEMENTS OF C CURRENTLY IN USE
          C   NMAX = SIZE OF C
          C   K = SUBSCRIPT OF SPARSE ARRAY TO LOOK UP
          C   STORE = .TRUE. IF ZEROS STORED EXPLICITLY, ELSE, FALSE.
          C   NONE = .FALSE. IF ZEROS NOT STORED OR ZEROS STORED AND
          C   ELEMENT IS FOUND IN THE STORED ARRAY
          C   .TRUE. IF ZEROS ARE STORED AND THE ELEMENT IS
          C   NOT FOUND (OUTPUT QUANTITY)
          C
          C PROGRAMMER: ROBERT H. FRENCH
          C DATE: 11 JANUARY 1984
          C
          0002      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          0003      VIRTUAL ICSUB(NMAX), C(NMAX)
          0004      LOGICAL*1 STORE, NONE
          0005      NONE=.FALSE.
          0006      DO 10 I=1,N
          0007      IF(ICSUB(I).NE.K)GOTO 10
          0008      COUNT=C(I)
          0009      10      CONTINUE
          0010      RETURN
          0011      IF(STORE) THEN
          0012      NONE=.TRUE..
          0013      ELSE
          0014      COUNT=0.
          0015      END IF
          0016      RETURN
          0017      END
```

CLIP13SGR.FTN;6 FORTRAN-77 V4.0-1 /F77/WR 10:44:54 16-Jul-86

PDP-11 FORTRAN-77 V4.0-1 /F77/WR Page 16 CLPL3SGR.FTN;6

SUBROUTINE LICM (NMAX, LLOW, UPP, ISUBN, R, MEAR)
8001

```
C THIS SUBROUTINE COMPUTES THE EQUIVALENT LINEAR SUBSCRIPT FOR
C A MULTIDIMENSIONAL ARRAY OF NDIM DIMENSIONS
C
C IF THE ARRAY A IS DEFINED AS
C
C      DIMENSION A(1:NDIM1):1:UP(1) ... :1:UP(NDIM) :1:UP(NDIM)
C      AND ISUB(1),...,ISUB(NDIM) IS A SET OF SUBSCRIPTS FOR A,
C      THEN THIS SUBROUTINE RETURNS IN LINEAR THE OFFSET FROM THE
C      ORIGIN OF A TO THE ELEMENT A(ISUB(1),...,ISUB(NDIM)). ASSUMING
C      THE FIRST SUBSCRIPT VARIES MOST RAPIDLY.
C
C USAGE:
```

```

C DIMENSION IL0W(NDIM), IUP(NDIM), ISUB(NDIM)
C DATA IL0W/low/lower limits of defined subscripts of array/
C DATA IUP/upper limits of defined subscripts of array/
C C SET ISUB TO DESIRED SEQUENCE OF SUBSCRIPTS...
C CALL LOCN(NDIM, IL0W, IUP, ISUB, LINEAR)

C WHERE NDIM = NUMBER OF DIMENSIONS THE ARRAY HAS
C       IL0W = ARRAY OF LOWER SUBSCRIPT BOUNDS;
C       IUP = ARRAY OF UPPER SUBSCRIPT BOUNDS;
C       ISUB = ARRAY CONTAINING SUBSCRIPT FOR WHICH LOCATION IS
C              TO BE COMPUTED
C       LINEAR = RETURNED VALUE OF OFFSET INTO ARRAY IN MEMORY

```

```

C PROGRAMMER: ROBERT H. FRENCH
C DATE: 11 JANUARY 1984
C
C DIMENSION ILOW(NDIM), IUP(NDIM), ISUB(NDIM)
C LINEAR=0
C DO 10 I=1,NDIM-1
C     J=NDIM-I+1
C     LINEAR=(LINEAR+(ISUB(I)-ILOW(J)))*(IUP(J-1)-ILOW(J-1)+1)
C     CONTINUE
C     LINEAR=LINEAR+ISUB(1)-ILOW(1)
C 10    RETURN
C END

```

```

C LMAX = NUMBER OF LOOPS NESTED
C GO = LOGICAL VARIABLE - TRUE, IF JUMP BACK TO BEGINNING OF
C STATEMENTS IN THE RANGE OF THE LOOP SHOULD OCCUR.
C .FALSE. OTHERWISE (I.E. OUTER-MOST LOOP TERMINATED)
C
C PROGRAMMER: ROBERT H. FRENCH          DATE: 11 JANUARY 1984
C
C DIMENSION LVEC(LMAX),LLON(LMAX)
C DO 1 N=1,LMAX
C     LVEC(N)=LLON(N)
C 1 CONTINUE
C     RETURN
C     END

```

104

POP-11 FORTRAN-77 V4.0-1 10:44:58 16-Jul-86 Page 17
 CL1PL3SGR.FTN:6 /F77/NR
 SUBROUTINE VLITER(LVEC,LL0N,LUP,LINC,LMAX,GO)
 C LOOP ITERATION LOGIC FOR A "VECTOR DO-LOOP"
 C SEE DETAILED COMMENTS IN SUBROUTINE VLINIT FOR USAGE AND
 C PARAMETER DEFINITIONS
 C PROGRAMMER: ROBERT H. FRENCH
 C DATE: 11 JANUARY 1984
 C
 LOGICAL *1 GO
 DIMENSION LVEC(LMAX),LL0N(LMAX),LUP(LMAX),LINC(LMAX)
 DO 100 NDX=1,LMAX
 GO=.TRUE.
 DO 100 NDX=1,LMAX
 NSUB=LMAX+1-NDX
 LVEC(NSUB)=LVEC(NSUB)+LINC(NSUB)
 IF((LINC(NSUB).GE.0.AND.LVEC(NSUB).LE.LUP(NSUB))
 OR((LINC(NSUB).LT.0.AND.LVEC(NSUB).GE.LUP(NSUB))) RETURN
 LVEC(NSUB)=LL0N(NSUB)
 CONTINUE
 GO=.FALSE..
 RETURN
 END
 100 CONTINUE
 1000

PDP-11 FORTRAN-77 V4.0-1 10:45:00 16-Jul-86 Page 18
 CL1PL3SGR.FTN:6 /F77/NR
 SUBROUTINE PRINOP(1,KN,KQ,KN,AIN)
 C THIS SUBROUTINE COMPUTES THE EVENT PROBABILITY FOR ALL
 C POSSIBLE JAMMING PATTERNS WITH NON-ZERO PROBABILITY FOR
 C L=1 MODE/SYMBOL FOR BMFSK/FN IN PBWJ
 C
 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
 DIMENSION I(4)
 AIN=0.00
 KJAM=0
 DO 1 K=1,KN
 KJAM=KJAM+1(K)
 CONTINUE
 1 IF(KJAM.GT.NIN0(KQ,KH)) RETURN
 C IF THIS IS AN IMPOSSIBLE CASE, RETURN WITH RESULT = 0.0
 KPMAX=KJAM-1
 LPMAX=KN-KJAM-1
 JPMAX=KH-1
 IMAX=MAX0(KPMAX,LPMAX,JPMAX)
 PROD=1.00
 0009
 0010
 0011
 0012
 0013
 0014
 0015
 0016
 0017
 0018
 0019
 0020
 0021
 0022
 0023
 0024
 0025
 0026

```

PDP-11 FORTRAN-77 V4.0-1 10:45:03 16-Jul-86 Page 19
CLIP13SGF.FTN;6 /F77/MR

0001 BLOCK DATA
      C INITIALIZE SHARED CONSTANTS
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /SHARE1/ DSMR(3,4)
      COMMON /SHARE2/ LOM(4),LINEC(4)
      COMMON /WTS/ X(5),W(5)

      C WEIGHTS AND ABSISSAS FOR 10-POINT GAUSSIAN QUADRATURE
      C
      DATA X/ 0.14887433998163100,
      S   0.41339539412924700,
      S   0.679403968829902400,
      S   0.86506336668898500,
      S   0.97390652851717200 /
      DATA W/ 0.29552422471475300,
      S   0.26926671930999600,
      S   0.21908636251598200,
      S   0.14945134915058100,
      S   0.066671344303888800 /
      S   0.048671344303888800 /
      C DEFAULT LISTS FOR INTERACTIVE PARAMETER INPUTS
      C ARE SHARED WITH LARGE WORKING STORAGE ARRAYS SINCE THEY
      C MAY BE DESTROYED ONCE THE INPUT PARAMETERS ARE SET UP
      DATA DSMR /13.3524700, 12.313300, 10.944300,
      S   10.60637200, 9.628400, 8.3524500,
      S   9.0940100, 8.169000, 6.97199500,
      S   8.078500, 7.199600, 6.06964600/
      C FREQUENTLY NEEDED CONSTANT ARRAYS AND SCALARS
      DATA LOM/4*0/,LINEC/4*1/
      END

```

```

PDP-11 FORTRAN-77 V4.0-1 10:45:05 16-Jul-86 Page 20
CLIP13SGF.FTN;6 /F77/MR

0001 SUBROUTINE ADQUAD(XL,XU,Y,QR,F,TOL,WORK,STACK,HEAP,N,KODE)
      C ADAPTIVE QUADRATURE ALGORITHM
      C XL - LOWER LIMIT OF INTEGRAL (IN)
      C XU - UPPER LIMIT OF INTEGRAL (IN)
      C Y - VALUE OF INTEGRAL (OUT)
      C QR - NAME OF A QUADRATURE RULE SUBROUTINE (IN)
      C WITH CALLING SEQUENCE
      CALL QR(XL,XU,F,Y)
      C F - NAME OF FUNCTION TO BE INTEGRATED (IN)
      C TOL - ERROR TOLERANCE FOR FINAL ANSWER (IN)
      C WORK - WORK ARRAY OF SIZE N (IN)
      C STACK - SECOND WORK ARRAY OF SIZE N, MUST NOT BE
      C HEAP- THIRD WORK ARRAY, SIZE N, DISTINCT FROM WORK AND STACK
      C SAME ARRAY AS WORK (IN)
      C N - SIZE OF WORK AND STACK; MAX. NO. OF BISECTIONS (IN)
      C KODE - ERROR INDICATOR (OUT)
      C 0 -- NO ERROR
      C 1 -- WORK ARRAYS TOO SMALL
      C 2 -- EPS DIVIDED TO ZERO, EITHER ASKING FOR TOO
      C TIGHT A TOLERANCE OR ROUND-OFF PREVENTS
      C ATTAINING REQUIRED ACCURACY
      C R. H. FRENCH, 14 AUGUST 1984
      C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      EXTERNAL F
      DIMENSION WORK(N),STACK(N),HEAP(N)
      KODE=0
      0002 Y=0.00
      WORK(1)=XU
      0003 CALL QR(XL,XU,F,T)
      0004 HEAP(1)=T
      0005 A=XL
      0006 NPPTS=1
      0007 EPS=TOL
      0008 CALL QR(XL,XU,F,T)
      0009 0009
      0010 A=XL
      0011 NPPTS=1
      0012 EPS=EPS
      0013 STACK(1)=EPS
      0014 B=WORK(NPPTS)
      10 0014
      XM=(A+B)*0.500
      0015 CALL QR(A,XM,F,P1)
      0016 CALL QR(XM,B,F,P2)
      0017 0017
      0018 IF(DABS(T-P1-P2).LE.EPS) GOTO 20
      C SPLIT IT
      0019 NPPTS=NPPTS+1
      0020 IF(NPPTS.GT.N) THEN
      0021 KODE=1
      0022 RETURN
      END IF
      0023 WORK(NPPTS)=XM
      0024 HEAP(NPPTS)=P2
      0025 T=P1
      0026 EPS=EPS/2.00
      0027

```

PDP-11 FORTRAN-77 V4.0-1 10:45:05 16-Jul-86 Page 21
 CL1PL3SSR.FTN;6 /F77/WR

```

0028 IF(EPS.EQ.0.00) THEN
0029   KODE=2
0030   RETURN
0031   END IF
0032   STACK(NPTS)=EPS
0033   GOTO 10
0034 C FINISHED A PIECE
0035   Y=Y+P1+P2
0036   EPS=STACK(NPTS)
0037   T=HEAP(NPTS)
0038   NPTS=NPTS-1
0039   A=8
0040   IF(NPTS.EQ.0) RETURN
0041   GOTO 10
0042 C

```

PDP-11 FORTRAN-77 V4.0-1 10:45:08 16-Jul-86 Page 22
 CL1PL3SSR.FTN;6 /F77/WR

```

0001      SUBROUTINE ADQUA2(XL,XU,Y,QR,F,TOL,WORK,STACK,HEAP,N,KODE)

C ADAPTIVE QUADRATURE ALGORITHM
C XL - LOWER LIMIT OF INTEGRAL (IN)
C XU - UPPER LIMIT OF INTEGRAL (IN)
C Y - VALUE OF INTEGRAL (OUT)
C QR - NAME OF A QUADRATURE RULE SUBROUTINE (IN)
C F - NAME OF FUNCTION TO BE INTEGRATED (IN)
C TOL - ERROR TOLERANCE FOR FINAL ANSWER (IN)
C WORK - WORK ARRAY OF SIZE N (IN)
C STACK - SECOND WORK ARRAY OF SIZE N, MUST NOT BE
C HEAP - THIRD WORK ARRAY, SIZE N, DISTINCT FROM WORK AND STACK
C SAME ARRAY AS WORK (IN)
C N - SIZE OF WORK AND STACK; MAX. NO. OF BISECTIONS (IN)
C KODE - ERROR INDICATOR (OUT)
C 0 -- NO ERROR
C 1 -- WORK ARRAYS TOO SMALL
C 2 -- EPS DIVIDED TO ZERO, EITHER ASKING FOR TOO
C      TIGHT A TOLERANCE OR ROUND-OFF PREVENTS
C      ATTAINING REQUIRED ACCURACY
C R. H. FRENCH, 14 AUGUST 1984
C
0002      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003      EXTERNAL F
0004      DIMENSION WORK(N),STACK(N),HEAP(N)
0005      KODE=0
0006      Y=0.D0
0007      WORK(1)=XU
0008      CALL QR(XL,XU,F,T)
0009      HEAP(1)=T
0010      A=XL
0011      NPTS=1
0012      EPS=TOL
0013      STACK(1)=EPS
0014      B=DRK(NPTS)
10      XM=(A+B)*0.5D0
0015      CALL QR(A,XM,F,P1)
0016      CALL QR(XM,B,F,P2)
0017      IF(DABS(T-P1-P2).LE.EPS) GOTO 20
0018      C SPLIT IT
0019      NPTS=NPTS+1
0020      IF(NPTS.GT.N) THEN
0021        KODE=1
0022        RETURN
0023      END IF
0024      WORK(NPTS)=XM
0025      HEAP(NPTS)=P2
0026      T=P1
0027      EPS=EPS/2.00

```

PDP-11 FORTRAN-77 V4.0-1 10-45:08
 CLIPL3SGR.FTN.6 /F77/MR

```

0028      IF(EPS.EQ.0.0D) THEN
0029        KODE=2
0030        RETURN
0031      END IF
0032      STACK(NPTS)=EPS
0033      GOTO 10
0034      C FINISHED A PIECE
0035      Y=Y+P1+P2
0036      EPS=STACK(NPTS)
0037      I=HEAP(NPTS)
0038      NPTS=NPTS-1
0039      A=B
0040      IF(NPTS.EQ.0) RETURN
0041      GOTO 10
END

```

Page 23

Page 24

PDP-11 FORTRAN-77 V4.0-1 10-45:12
 CLIPL3SGR.FTN.6 /F77/MR

```

0001      C 10-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL
0002      C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4
0003      C R. H. FRENCH, 28 FEBRUARY 1986
0004      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0005      COMMON /WTS/ X(5),W(5)
0006      ANSWER=0.00
0007      BM402=(B-A)/2.00
0008      BP402=(B+A)/2.00
0009      DO 10 J=1,5
0010      C=X(J)*BM402
0011      Y1=BP402+C
0012      Y2=BP402-C
0013      ANSWER=ANSWER+W(J)*(F(Y1)+F(Y2))
0014      CONTINUE
0015      ANSWER=ANSWER*BM402
0016      RETURN
END

```

0001 SUBROUTINE ADQUA3(XL,XU,Y,QR,F,TOL,WORK,STACK,HEAP,N,KODE)

```

C ADAPTIVE QUADRATURE ALGORITHM
C XL - LOWER LIMIT OF INTEGRAL (IN)
C XU - UPPER LIMIT OF INTEGRAL (IN)
C Y - VALUE OF INTEGRAL (OUT)
C QR - NAME OF A QUADRATURE RULE SUBROUTINE (IN)
C          WITH CALLING SEQUENCE
C          CALL QR(XL,XU,F,Y)
C F - NAME OF FUNCTION TO BE INTEGRATED (IN)
C TOL - ERROR TOLERANCE FOR FINAL ANSWER (IN)
C WORK - WORK ARRAY OF SIZE N (IN)
C STACK - SECOND WORK ARRAY OF SIZE N. MUST NOT BE
C SAME ARRAY AS WORK (IN)
C HEAP - THIRD WORK ARRAY, SIZE N, DISTINCT FROM WORK AND STACK
C N - SIZE OF WORK AND STACK; MAX. NO. OF BISECTIONS (IN)
C KODE - ERROR INDICATOR (OUT)
C 0 -- NO ERROR
C 1 -- WORK ARRAYS TOO SMALL
C 2 -- EPS DIVIDED TO ZERO, EITHER ASKING FOR TOO
C      TIGHT A TOLERANCE OR ROUND-OFF PREVENTS
C      ATTAINING REQUIRED ACCURACY
C R. H. FRENCH, 14 AUGUST 1984
C
C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C EXTERNAL F
C DIMENSION WORK(N),STACK(N),HEAP(N)
C KODE=0
C Y=0.D0
C WORK(1)=XU
C CALL QR(XL,XU,F,T)
C HEAP(1)=T
C A=XL
C NPTS=1
C EPS=TOL
C STACK(1)=EPS
C 10 B=WORK(NPTS)
C XM=(A+B)*0.5D0
C CALL QR(A,XM,F,P1)
C CALL QR(XM,B,F,P2)
C *** MAKE IT A RELATIVE TEST FOR THIS INTEGRAL ***
C IF(DABS(T-P1).LE.DABST(T-EPS)) GOTO 20
C SPLIT IT
C NPTS=NPTS+1
C IF(NPTS.GT.N) THEN
C KODE=1
C RETURN
C END IF
C WORK(NPTS)=XM
C HEAP(NPTS)=P2
C T=P1
C EPS=EPS/2. DO

```

PDP-11 FORTRAN-77 V4.0-1 10-45:17 16-Jul-86

CLIP13SER.FTN:6 /F77/M*

Page 27

PDP-11 FORTRAN-77 V4.0-1 10-45:19 16-Jul-86
CLIP13SER.FTN:6 /F77/M*

Page 28

SUBROUTINE DGTE(N,A,B,F,ANSWER)

C 10-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL

C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4

C R. H. FRENCH, 28 FEBRUARY 1986

C IMPLICIT DOUBLE PRECISION (A-H,O-Z)

COMMON /WTS/ X(5),W(5)

ANSWER=0.00

BMA02=(B-A)/2.00

BPA02=(B+A)/2.00

00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015

00001 C 10-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL

C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4

C R. H. FRENCH, 28 FEBRUARY 1986

C IMPLICIT DOUBLE PRECISION (A-H,O-Z)

COMMON /WTS/ X(5),W(5)

ANSWER=0.00

BMA02=(B-A)/2.00

BPA02=(B+A)/2.00

00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015

00001 C 10-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL

C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4

C R. H. FRENCH, 28 FEBRUARY 1986

C IMPLICIT DOUBLE PRECISION (A-H,O-Z)

COMMON /WTS/ X(5),W(5)

ANSWER=0.00

BMA02=(B-A)/2.00

BPA02=(B+A)/2.00

00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015

00001 C 10-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL

C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4

C R. H. FRENCH, 28 FEBRUARY 1986

C IMPLICIT DOUBLE PRECISION (A-H,O-Z)

COMMON /WTS/ X(5),W(5)

ANSWER=0.00

BMA02=(B-A)/2.00

BPA02=(B+A)/2.00

00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015

00001 C 10-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL

C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4

C R. H. FRENCH, 28 FEBRUARY 1986

C IMPLICIT DOUBLE PRECISION (A-H,O-Z)

COMMON /WTS/ X(5),W(5)

ANSWER=0.00

BMA02=(B-A)/2.00

BPA02=(B+A)/2.00

00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015

00001 C 10-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL

C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4

C R. H. FRENCH, 28 FEBRUARY 1986

C IMPLICIT DOUBLE PRECISION (A-H,O-Z)

COMMON /WTS/ X(5),W(5)

ANSWER=0.00

BMA02=(B-A)/2.00

BPA02=(B+A)/2.00

00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015

00001 C 10-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL

C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4

C R. H. FRENCH, 28 FEBRUARY 1986

C IMPLICIT DOUBLE PRECISION (A-H,O-Z)

COMMON /WTS/ X(5),W(5)

ANSWER=0.00

BMA02=(B-A)/2.00

BPA02=(B+A)/2.00

00001 00002 00003 00004 00005 00006 00007 00008 00009 00010 00011 00012 00013 00014 00015

PDP-11 FORTRAN-77 V4.0-1 /F77/MR 10:45:21 16-Jul-86 Page 29

PDP-11 FORTRAN-77 V4.0-1 /F77/MR 10:45:21 16-Jul-86 Page 30

```

0001      SUBROUTINE MINSER(F,FMIN,XMIN,STEP,GUESS,BLIM,ULIM,TOL)
          C SEARCH FOR MINIMUM OF F(X) OVER THE INTERVAL BLIM <= X <= ULIM
          C TROUBLE MAY OCCUR IF F(X) HAS MULTIPLE LOCAL MINIMA WITHIN THE
          C SEARCH INTERVAL OR IF THE FUNCTION IS VERY STEEP AND STEP IS
          C TOO BIG.
          C
          C F = NAME OF FUNCTION TO BE MINIMIZED
          C FMIN = MINIMUM VALUE OF F(X) OVER INTERVAL
          C XMIN = VALUE OF X FOR WHICH FMIN OCCURS
          C STEP = INITIAL STEP SIZE FOR SEARCH
          C GUESS = INITIAL GUESS AT XMIN, BLIM <= GUESS <= ULIM
          C BLIM = LOWER LIMIT OF SEARCH INTERVAL
          C ULIM = UPPER LIMIT OF SEARCH INTERVAL
          C TN = TOLERANCE ON XMIN; SEARCH STOPS WHEN DX < TOL
          C
          C NOTE: F MUST BE A DOUBLE PRECISION FUNCTION OF ONE DOUBLE PRECISION
          C ARGUMENT. ANY PARAMETERS CAN BE PASSED FROM THE CALLER VIA
          C A COMMON BLOCK.
          C
          C PROGRAMMER: ROBERT H. FRENCH   DATE: 17 MARCH 1986
          C
          C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          C
          C 0002  0003  X=GUESS
          C 0004  0005  SLMAX=DABS(X-BLIM)
          C 0006  0007  SUMAX=DABS(ULIM-X)
          C 0008  0009  DX=DMIN1(STEP,SLMAX,SUMAX)
          C 0010  0011  TEST=TOL
          C 0012  0013  10  F0=F(X)
          C 0014  0015  F1=F(X+DX)
          C 0016  0017  C ARE WE GOING IN THE RIGHT DIRECTION?
          C 0018  0019  C ELSE WE MUST BE CLOSE TO A MIN. AT X=GUESS, SO CUT
          C 0020  0021  C STEP SIZE AND TRY AGAIN
          C 0022  0023  11  IF(F1.LE.F0) GOTO 100
          C 0024  0025  IF(DABS(DX).GE.TEST) GOTO 10
          C 0026  0027  C CLOSE ENOUGH AT GUESS
          C 0028  0029  12  XMIN=X
          C 0030  0031  FMIN=F0
          C 0032  0033  RETURN
          C 0034  0035  C MIN AT X=ULIM
          C 0036  0037  XMIN=BLIM
          C 0038  0039  C HAVE PASSED MIN. IS IT LOCATED CLOSELY ENOUGH YET?
          C 0040  0041  13  IF(DABS(DX).LE.TEST) GOTO 300
          C 0042  0043  C ... NO, CUT STEP SIZE AND TRY AGAIN
          C 0044  0045  14  IF(F0 >= F1 & F2>= F1 AND ABS(DX)<MIN. DX, CALL F1 THE MIN.
          C 0046  0047  15  FMIN=F1
          C 0048  0049  RETURN
          C 0050  0051  C NOW GOING RIGHT DIRECTION.
          C 0052  0053  C KEEP GOING UNTIL PAST MINIMUM BY ONE STEP.
          C 0054  0055  16  X2=X+DX
          C 0056  0057  C HAVE WE REACHED END POINT?
          C 0058  0059  17  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0060  0061  18  FMIN=X
          C 0062  0063  RETURN
          C 0064  0065  19  X2=X+DX
          C 0066  0067  C HAVE WE REACHED END POINT?
          C 0068  0069  20  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0070  0071  21  FMIN=X
          C 0072  0073  RETURN
          C 0074  0075  22  X2=X+DX
          C 0076  0077  C HAVE WE REACHED END POINT?
          C 0078  0079  23  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0080  0081  24  FMIN=X
          C 0082  0083  RETURN
          C 0084  0085  25  X2=X+DX
          C 0086  0087  C HAVE WE REACHED END POINT?
          C 0088  0089  26  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0090  0091  27  FMIN=X
          C 0092  0093  RETURN
          C 0094  0095  28  X2=X+DX
          C 0096  0097  C HAVE WE REACHED END POINT?
          C 0098  0099  29  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0100  0101  30  FMIN=X
          C 0102  0103  RETURN
          C 0104  0105  31  X2=X+DX
          C 0106  0107  C HAVE WE REACHED END POINT?
          C 0108  0109  32  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0110  0111  33  FMIN=X
          C 0112  0113  RETURN
          C 0114  0115  34  X2=X+DX
          C 0116  0117  C HAVE WE REACHED END POINT?
          C 0118  0119  35  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0120  0121  36  FMIN=X
          C 0122  0123  RETURN
          C 0124  0125  37  X2=X+DX
          C 0126  0127  C HAVE WE REACHED END POINT?
          C 0128  0129  38  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0130  0131  39  FMIN=X
          C 0132  0133  RETURN
          C 0134  0135  40  X2=X+DX
          C 0136  0137  C HAVE WE REACHED END POINT?
          C 0138  0139  41  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0140  0141  42  FMIN=X
          C 0142  0143  RETURN
          C 0144  0145  43  X2=X+DX
          C 0146  0147  C HAVE WE REACHED END POINT?
          C 0148  0149  44  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0150  0151  45  FMIN=X
          C 0152  0153  RETURN
          C 0154  0155  46  X2=X+DX
          C 0156  0157  C HAVE WE REACHED END POINT?
          C 0158  0159  47  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0160  0161  48  FMIN=X
          C 0162  0163  RETURN
          C 0164  0165  49  X2=X+DX
          C 0166  0167  C HAVE WE REACHED END POINT?
          C 0168  0169  50  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0170  0171  51  FMIN=X
          C 0172  0173  RETURN
          C 0174  0175  52  X2=X+DX
          C 0176  0177  C HAVE WE REACHED END POINT?
          C 0178  0179  53  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0180  0181  54  FMIN=X
          C 0182  0183  RETURN
          C 0184  0185  55  X2=X+DX
          C 0186  0187  C HAVE WE REACHED END POINT?
          C 0188  0189  56  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0190  0191  57  FMIN=X
          C 0192  0193  RETURN
          C 0194  0195  58  X2=X+DX
          C 0196  0197  C HAVE WE REACHED END POINT?
          C 0198  0199  59  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0200  0201  60  FMIN=X
          C 0202  0203  RETURN
          C 0204  0205  61  X2=X+DX
          C 0206  0207  C HAVE WE REACHED END POINT?
          C 0208  0209  62  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0210  0211  63  FMIN=X
          C 0212  0213  RETURN
          C 0214  0215  64  X2=X+DX
          C 0216  0217  C HAVE WE REACHED END POINT?
          C 0218  0219  65  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0220  0221  66  FMIN=X
          C 0222  0223  RETURN
          C 0224  0225  67  X2=X+DX
          C 0226  0227  C HAVE WE REACHED END POINT?
          C 0228  0229  68  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0230  0231  69  FMIN=X
          C 0232  0233  RETURN
          C 0234  0235  70  X2=X+DX
          C 0236  0237  C HAVE WE REACHED END POINT?
          C 0238  0239  71  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0240  0241  72  FMIN=X
          C 0242  0243  RETURN
          C 0244  0245  73  X2=X+DX
          C 0246  0247  C HAVE WE REACHED END POINT?
          C 0248  0249  74  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0250  0251  75  FMIN=X
          C 0252  0253  RETURN
          C 0254  0255  76  X2=X+DX
          C 0256  0257  C HAVE WE REACHED END POINT?
          C 0258  0259  77  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0260  0261  78  FMIN=X
          C 0262  0263  RETURN
          C 0264  0265  79  X2=X+DX
          C 0266  0267  C HAVE WE REACHED END POINT?
          C 0268  0269  80  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0270  0271  81  FMIN=X
          C 0272  0273  RETURN
          C 0274  0275  82  X2=X+DX
          C 0276  0277  C HAVE WE REACHED END POINT?
          C 0278  0279  83  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0280  0281  84  FMIN=X
          C 0282  0283  RETURN
          C 0284  0285  85  X2=X+DX
          C 0286  0287  C HAVE WE REACHED END POINT?
          C 0288  0289  86  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0290  0291  87  FMIN=X
          C 0292  0293  RETURN
          C 0294  0295  88  X2=X+DX
          C 0296  0297  C HAVE WE REACHED END POINT?
          C 0298  0299  89  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0300  0301  90  FMIN=X
          C 0302  0303  RETURN
          C 0304  0305  91  X2=X+DX
          C 0306  0307  C HAVE WE REACHED END POINT?
          C 0308  0309  92  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0310  0311  93  FMIN=X
          C 0312  0313  RETURN
          C 0314  0315  94  X2=X+DX
          C 0316  0317  C HAVE WE REACHED END POINT?
          C 0318  0319  95  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0320  0321  96  FMIN=X
          C 0322  0323  RETURN
          C 0324  0325  97  X2=X+DX
          C 0326  0327  C HAVE WE REACHED END POINT?
          C 0328  0329  98  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0330  0331  99  FMIN=X
          C 0332  0333  RETURN
          C 0334  0335  100  X2=X+DX
          C 0336  0337  C HAVE WE REACHED END POINT?
          C 0338  0339  101  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0340  0341  102  FMIN=X
          C 0342  0343  RETURN
          C 0344  0345  103  X2=X+DX
          C 0346  0347  C HAVE WE REACHED END POINT?
          C 0348  0349  104  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0350  0351  105  FMIN=X
          C 0352  0353  RETURN
          C 0354  0355  106  X2=X+DX
          C 0356  0357  C HAVE WE REACHED END POINT?
          C 0358  0359  107  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0360  0361  108  FMIN=X
          C 0362  0363  RETURN
          C 0364  0365  109  X2=X+DX
          C 0366  0367  C HAVE WE REACHED END POINT?
          C 0368  0369  110  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0370  0371  111  FMIN=X
          C 0372  0373  RETURN
          C 0374  0375  112  X2=X+DX
          C 0376  0377  C HAVE WE REACHED END POINT?
          C 0378  0379  113  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0380  0381  114  FMIN=X
          C 0382  0383  RETURN
          C 0384  0385  115  X2=X+DX
          C 0386  0387  C HAVE WE REACHED END POINT?
          C 0388  0389  116  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0390  0391  117  FMIN=X
          C 0392  0393  RETURN
          C 0394  0395  118  X2=X+DX
          C 0396  0397  C HAVE WE REACHED END POINT?
          C 0398  0399  119  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0400  0401  120  FMIN=X
          C 0402  0403  RETURN
          C 0404  0405  121  X2=X+DX
          C 0406  0407  C HAVE WE REACHED END POINT?
          C 0408  0409  122  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0410  0411  123  FMIN=X
          C 0412  0413  RETURN
          C 0414  0415  124  X2=X+DX
          C 0416  0417  C HAVE WE REACHED END POINT?
          C 0418  0419  125  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0420  0421  126  FMIN=X
          C 0422  0423  RETURN
          C 0424  0425  127  X2=X+DX
          C 0426  0427  C HAVE WE REACHED END POINT?
          C 0428  0429  128  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0430  0431  129  FMIN=X
          C 0432  0433  RETURN
          C 0434  0435  130  X2=X+DX
          C 0436  0437  C HAVE WE REACHED END POINT?
          C 0438  0439  131  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0440  0441  132  FMIN=X
          C 0442  0443  RETURN
          C 0444  0445  133  X2=X+DX
          C 0446  0447  C HAVE WE REACHED END POINT?
          C 0448  0449  134  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0450  0451  135  FMIN=X
          C 0452  0453  RETURN
          C 0454  0455  136  X2=X+DX
          C 0456  0457  C HAVE WE REACHED END POINT?
          C 0458  0459  137  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0460  0461  138  FMIN=X
          C 0462  0463  RETURN
          C 0464  0465  139  X2=X+DX
          C 0466  0467  C HAVE WE REACHED END POINT?
          C 0468  0469  140  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0470  0471  141  FMIN=X
          C 0472  0473  RETURN
          C 0474  0475  142  X2=X+DX
          C 0476  0477  C HAVE WE REACHED END POINT?
          C 0478  0479  143  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0480  0481  144  FMIN=X
          C 0482  0483  RETURN
          C 0484  0485  145  X2=X+DX
          C 0486  0487  C HAVE WE REACHED END POINT?
          C 0488  0489  146  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0490  0491  147  FMIN=X
          C 0492  0493  RETURN
          C 0494  0495  148  X2=X+DX
          C 0496  0497  C HAVE WE REACHED END POINT?
          C 0498  0499  149  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0500  0501  150  FMIN=X
          C 0502  0503  RETURN
          C 0504  0505  151  X2=X+DX
          C 0506  0507  C HAVE WE REACHED END POINT?
          C 0508  0509  152  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0510  0511  153  FMIN=X
          C 0512  0513  RETURN
          C 0514  0515  154  X2=X+DX
          C 0516  0517  C HAVE WE REACHED END POINT?
          C 0518  0519  155  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0520  0521  156  FMIN=X
          C 0522  0523  RETURN
          C 0524  0525  157  X2=X+DX
          C 0526  0527  C HAVE WE REACHED END POINT?
          C 0528  0529  158  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0530  0531  159  FMIN=X
          C 0532  0533  RETURN
          C 0534  0535  160  X2=X+DX
          C 0536  0537  C HAVE WE REACHED END POINT?
          C 0538  0539  161  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0540  0541  162  FMIN=X
          C 0542  0543  RETURN
          C 0544  0545  163  X2=X+DX
          C 0546  0547  C HAVE WE REACHED END POINT?
          C 0548  0549  164  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0550  0551  165  FMIN=X
          C 0552  0553  RETURN
          C 0554  0555  166  X2=X+DX
          C 0556  0557  C HAVE WE REACHED END POINT?
          C 0558  0559  167  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0560  0561  168  FMIN=X
          C 0562  0563  RETURN
          C 0564  0565  169  X2=X+DX
          C 0566  0567  C HAVE WE REACHED END POINT?
          C 0568  0569  170  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0570  0571  171  FMIN=X
          C 0572  0573  RETURN
          C 0574  0575  172  X2=X+DX
          C 0576  0577  C HAVE WE REACHED END POINT?
          C 0578  0579  173  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0580  0581  174  FMIN=X
          C 0582  0583  RETURN
          C 0584  0585  175  X2=X+DX
          C 0586  0587  C HAVE WE REACHED END POINT?
          C 0588  0589  176  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0590  0591  177  FMIN=X
          C 0592  0593  RETURN
          C 0594  0595  178  X2=X+DX
          C 0596  0597  C HAVE WE REACHED END POINT?
          C 0598  0599  179  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0600  0601  180  FMIN=X
          C 0602  0603  RETURN
          C 0604  0605  181  X2=X+DX
          C 0606  0607  C HAVE WE REACHED END POINT?
          C 0608  0609  182  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0610  0611  183  FMIN=X
          C 0612  0613  RETURN
          C 0614  0615  184  X2=X+DX
          C 0616  0617  C HAVE WE REACHED END POINT?
          C 0618  0619  185  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0620  0621  186  FMIN=X
          C 0622  0623  RETURN
          C 0624  0625  187  X2=X+DX
          C 0626  0627  C HAVE WE REACHED END POINT?
          C 0628  0629  188  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0630  0631  189  FMIN=X
          C 0632  0633  RETURN
          C 0634  0635  190  X2=X+DX
          C 0636  0637  C HAVE WE REACHED END POINT?
          C 0638  0639  191  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0640  0641  192  FMIN=X
          C 0642  0643  RETURN
          C 0644  0645  193  X2=X+DX
          C 0646  0647  C HAVE WE REACHED END POINT?
          C 0648  0649  194  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0650  0651  195  FMIN=X
          C 0652  0653  RETURN
          C 0654  0655  196  X2=X+DX
          C 0656  0657  C HAVE WE REACHED END POINT?
          C 0658  0659  197  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0660  0661  198  FMIN=X
          C 0662  0663  RETURN
          C 0664  0665  199  X2=X+DX
          C 0666  0667  C HAVE WE REACHED END POINT?
          C 0668  0669  200  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0670  0671  201  FMIN=X
          C 0672  0673  RETURN
          C 0674  0675  202  X2=X+DX
          C 0676  0677  C HAVE WE REACHED END POINT?
          C 0678  0679  203  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0680  0681  204  FMIN=X
          C 0682  0683  RETURN
          C 0684  0685  205  X2=X+DX
          C 0686  0687  C HAVE WE REACHED END POINT?
          C 0688  0689  206  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0690  0691  207  FMIN=X
          C 0692  0693  RETURN
          C 0694  0695  208  X2=X+DX
          C 0696  0697  C HAVE WE REACHED END POINT?
          C 0698  0699  209  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0700  0701  210  FMIN=X
          C 0702  0703  RETURN
          C 0704  0705  211  X2=X+DX
          C 0706  0707  C HAVE WE REACHED END POINT?
          C 0708  0709  212  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0710  0711  213  FMIN=X
          C 0712  0713  RETURN
          C 0714  0715  214  X2=X+DX
          C 0716  0717  C HAVE WE REACHED END POINT?
          C 0718  0719  215  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0720  0721  216  FMIN=X
          C 0722  0723  RETURN
          C 0724  0725  217  X2=X+DX
          C 0726  0727  C HAVE WE REACHED END POINT?
          C 0728  0729  218  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0730  0731  219  FMIN=X
          C 0732  0733  RETURN
          C 0734  0735  220  X2=X+DX
          C 0736  0737  C HAVE WE REACHED END POINT?
          C 0738  0739  221  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0740  0741  222  FMIN=X
          C 0742  0743  RETURN
          C 0744  0745  223  X2=X+DX
          C 0746  0747  C HAVE WE REACHED END POINT?
          C 0748  0749  224  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0750  0751  225  FMIN=X
          C 0752  0753  RETURN
          C 0754  0755  226  X2=X+DX
          C 0756  0757  C HAVE WE REACHED END POINT?
          C 0758  0759  227  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0760  0761  228  FMIN=X
          C 0762  0763  RETURN
          C 0764  0765  229  X2=X+DX
          C 0766  0767  C HAVE WE REACHED END POINT?
          C 0768  0769  230  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0770  0771  231  FMIN=X
          C 0772  0773  RETURN
          C 0774  0775  232  X2=X+DX
          C 0776  0777  C HAVE WE REACHED END POINT?
          C 0778  0779  233  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0780  0781  234  FMIN=X
          C 0782  0783  RETURN
          C 0784  0785  235  X2=X+DX
          C 0786  0787  C HAVE WE REACHED END POINT?
          C 0788  0789  236  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0790  0791  237  FMIN=X
          C 0792  0793  RETURN
          C 0794  0795  238  X2=X+DX
          C 0796  0797  C HAVE WE REACHED END POINT?
          C 0798  0799  239  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0800  0801  240  FMIN=X
          C 0802  0803  RETURN
          C 0804  0805  241  X2=X+DX
          C 0806  0807  C HAVE WE REACHED END POINT?
          C 0808  0809  242  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0810  0811  243  FMIN=X
          C 0812  0813  RETURN
          C 0814  0815  244  X2=X+DX
          C 0816  0817  C HAVE WE REACHED END POINT?
          C 0818  0819  245  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0820  0821  246  FMIN=X
          C 0822  0823  RETURN
          C 0824  0825  247  X2=X+DX
          C 0826  0827  C HAVE WE REACHED END POINT?
          C 0828  0829  248  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0830  0831  249  FMIN=X
          C 0832  0833  RETURN
          C 0834  0835  250  X2=X+DX
          C 0836  0837  C HAVE WE REACHED END POINT?
          C 0838  0839  251  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0840  0841  252  FMIN=X
          C 0842  0843  RETURN
          C 0844  0845  253  X2=X+DX
          C 0846  0847  C HAVE WE REACHED END POINT?
          C 0848  0849  254  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0850  0851  255  FMIN=X
          C 0852  0853  RETURN
          C 0854  0855  256  X2=X+DX
          C 0856  0857  C HAVE WE REACHED END POINT?
          C 0858  0859  257  IF(X2.GT.ULIM .OR. X2.LT.BLIM) GOTO 110
          C 0860  0861  258  FMIN=X
          C 0862  0863  RETURN
```

PPF-11 FORTRAN-77 V4.0-1 10:45:25 16-Jul-86

/F77/WR
CLJPL3SGR.FTN;6

Page 31

PPF-11 FORTRAN-77 V4.0-1 10:45:25 16-Jul-86

/F77/WR
CLJPL3SGR.FTN;6

Page 32

```
0001      SUBROUTINE PSEL(JSUB,MM,PROB)
C RANDOM MFSK/FH IN PARTIAL BAND NOISE JAMMING,
C GIVEN A JAMMING EVENT, WITH CLIPPER RECEIVER
C JSUB - JAMMING EVENT VECTOR
C LL - NUMBER OF HOPS/SYMBOL
C MM - ALPHABET SIZE
C PROB - RESULTING CONDITIONAL ERROR PROBABILITY
C
C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
DIMENSION JSUB(MM), WORK(30), HEAF(30)
EXTERNAL DG10, PRAND
INTEGER NCHAN(0:3)
COMMON /JAMCLT/ NCHAN
COMMON /DENPAR/ BIGK, AAB, BAB, LJAM, AAB2, BAB2, ABBAB,
              TAU, TAU2, TAU3, TAUK, TAUK2, TAUK3
COMMON /PARDEN/ RHON, RHOI
COMMON /QUES/ QO, Q1
COMMON /SCJAM/ JAMSC
KSUB=0
DO 6 1=1,MM
  KSUB=KSUB+JSUB(1)
CONTINUE
6  C SET UP VALUES WHICH WILL REMAIN IF THIS IS THE NOTHING-JAMMED CASE
  0015  IF(KSUB.NE.0) THEN
        0016    00=0.0*DSORT(0.500*RHON),DSORT(2.00*TAU)
        0017    TAUK=0.0
        0018    C IF ANYTHING IS JAMMED, SET UP JAMMING-RELATED QUANTITIES
        0019    IF(KSUB.NE.0) THEN
        0020      BIGK=RHON/RHOI
        0021      BK1=BIGK-1.0D0
        0022      AAB=BIGK/BK1
        0023      BAB=AAB*AAB
        0024      BAB2=BAB*BAB
        0025      BABBAB=BAB*BAB
        0026      TAUK=TAU/BIGK
        0027      TAUK2=TAU2/BIGK
        0028      TAUK3=TAU3/BIGK
        0029      Q1-Q(2.00*DSORT(0.500*RHON),DSQRT(2.00*TAUK))
        0030    END IF
C COUNT NUMBER OF NONSIGNAL CHANNELS WITH Lm HOPS JAMMED
C
0031    DO 10 I=0,3
        0032      NCHAN(I)=0
        0033    10  CONTINUE
        0034    DO 11 I=2,MM
        0035      KSUB=JSUB(I)
        0036      NCHAN(KSUB)=NCHAN(KSUB)+1
        0037    11  CONTINUE
        0038      JAMSC=JSUB(1)
```

POP-11 FORTRAN-77 V4.0-1
CL.IPL35GR.FTN:6 /F77/MR

Page 33

10:45:30
16-Jul-86

POP-11 FORTRAN-77 V4.0-1
CL.IPL35SR.FTN:6 /F77/MR

Page 36

```

0001    DOUBLE PRECISION FUNCTION PGRND(BETA)
0002    IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0003    INTEGER NCHAN(0:3)
0004    COMMON /SCJAM/ JAMSC
0005    COMMON /JAMCN/ NCHAN
0006    COMMON /DENPAR/ B1GK, AAB, BAB, LJAM, AAB2, BAB2, AABBA,
0007      COMMON /PARDEN/ RHON, RHOI,
0008      COMMON /QUES/ QD, Q1
0009      PROD=1. DO
0010      DO 10 I=0,3
0011      IF(NCHAN(I).NE.0) THEN
0012        LJAM=I
0013        X=G1(BETA)
0014        PROD=PROD-DX1(X,NCHAN(I))
0015      END IF
0016      CONTINUE
0017      LJAM=JAMSC
0018      Y=PZ1(BETA)
0019      PGRND=Y*PROD
0020      RETURN
0021      END

```

J-18
POP-11 FORTRAN-77 V4.0-1
CL.IPL35RP.FTN:6 /F77/MR

Page 34
10:45:33
16-Jul-86

```

0001    SUBROUTINE TEST(ID)
0002    C TEST RETURN CODE FROM BESSEL FUNCTION
0003    IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0004    COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
0005    IF(KODE.EQ.0) RETURN
0006    WRITE(5,1) KODE, ID
0007    1 FORMAT(' BESSEL FUNCTION CODE = ',ID,' FROM CALL NUMBER ',15)
0008    STOP 'FATAL ERROR.'
0009    END

```

POP-11 FORTRAN-77 V4.0-1
CL.IPL35SR.FTN:6 /F77/MR

Page 35
10:45:35
16-Jul-86

```

0001    SUBROUTINE TEST2(KODE, ID)
0002    C TEST RETURN CODE FROM ADQUAD/ADQUA2/ADQUA3
0003    IF(KODE.EQ.0) RETURN
0004    WRITE(5,1) KODE, 10
0005    1 FORMAT(' ADAPTIVE INTEGRATOR CODE = ',ID,' FROM CALL NUMBER ',15)
0006    STOP 'FATAL ERROR.'
0007    END

```

```

0001      C SIGNAL CHANNEL P.D.F. WITH CHANGE OF VARIABLE Y=AX
0002      IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0003      DIMENSION WORK1(30), STACK1(30), HEAP1(30)
0004      LOGICAL *1 REG1, REG2, REG3
0005      EXTERNAL DEX, F20A1, F30A2, F30B1,
0006      F30B2, F31A, F31B1, F31B2, F31B3, F31C2,
0007      F32A, F32B1, F32B2, F32B3, F32C2,
0008      F33A1, F33A2, F33B1, F33B2
0009      COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
0010      COMMON /DEMPAR/ AAB, BAB, LJAM, AAB2, BAB2, AABBA,
0011      TAU, TAU2, TAU3, TAUK, TAUK2, TAUK3
0012      COMMON /PARDEN/ RHON, RHOT,
0013      COMMON /QUES/ QD, Q1
0014      COMMON /XCOM/ XCOM
0015      COMMON /OUTER/ XXX, XXXX
0016      XXXY
0017      IF(LJAM.GE.1) THEN
0018        XXXX=Y/BIGK
0019        YK=XXXK
0020        YT1=(Y-TAU)/BIGK
0021        YT2=(Y-TAU2)/BIGK
0022        END IF
0023        REG1=Y.GE.0.00 .AND. Y.LT.TAU
0024        REG2=Y.GE.TAU .AND. Y.LT.TAU2
0025        REG3=Y.GE.TAU2 .AND. Y.LT.TAU3
0026        C THREE HOPS PER SYMBOL
0027        PZ1=Y/(3.00*RHON)*DEAP(BARG1-Y-3.00*RHON)+B1
0028        ELSE IF(REG2) THEN
0029          BARG1=DSQRT(8.00*RHON*Y)
0030          NO HOPS JAMMED
0031          IF(REG1) THEN
0032            101=2
0033            BARG1=DSQRT(12.00*RHON*Y)
0034            CALL DXBT(3101)
0035            101=1
0036            CALL DXBT(3100)
0037            PZ1=Y/(3.00*RHON)*DEAP(BARG1-Y-3.00*RHON)+B1
0038            ELSE IF(REG3) THEN
0039              BARG1=DSQRT(8.00*RHON*(Y-TAU))
0040              101=2
0041              CALL DXBT(3101)
0042              PZ1=Y/(3.00*DSQRT(2.00*(Y-TAU)/RHON))*DEAP(BARG1-Y-3.00*RHON)+B1
0043              XCOM=Y+3.00*RHON
0044              CALL ADQUA2(Y-TAU, TAU, ANSWER, DEX, F30A1, 1, D-9, WORK1, STACK1,
0045              HEAP1, 30, KOD)
0046              CALL TEST2(KOD, 3100)
0047              PART=PART+ANSWER/2.00

```

```

PDP-11 FORTRAN-77 V4.0-1          10:45:36   16-Jul-86
CL1PL3SGR.FTN.6      /F77/MR   Page 37

0037      CALL ADQUA2(TAU,Y,ANSWER,DGX,F30A2,1.0-9,WORK1,STACK1)
           HEAP1,30,K00)
0038      $     CALL TEST2(K00,3101)
           P21=PART+ANSWER
0039      $     ELSE IF (REG3) THEN
           BARG1=DQR(4.0*D*RHON*(Y-TAU2))
0040      $     101=0
           CALL DXBT((3103)
0041      $     PAR)=3.0*D*Q0*D*EXP(BARG1-Y+TAU2-RHON)*B1
0042      $     XCOM=Y-TAU+Z.D*D*RHON
           CALL ADQUA2(Y-TAU,TAU2,ANSWER,DGX,F30B1,1.0-9,WORK1,ST
           HEAP1,30,K00)
0043      $     CALL TEST2(K00,3103)
           PAR=P1+3.0*D*Q0*ANSWER
0044      $     XCOM=Y+3.0*D*RHON
           CALL ADQUA2(Y-TAU,TAU2,ANSWER,DGX,F30B2,1.0-9,WORK1,ST
           HEAP1,30,K00)
0045      $     CALL TEST2(K00,3104)
           P21=PART+ANSWER
0046      $     ELSE
           PZ1=0.0D
0047      $     END IF
           6010 9000
0048      $     C     ONE HOP JAMMED
0049      $     C     ONE HOP JAMMED
0050      $     C     ONE HOP JAMMED
0051      $     C     ONE HOP JAMMED
0052      $     C     ONE HOP JAMMED
0053      $     C     ONE HOP JAMMED
0054      $     C     ONE HOP JAMMED
0055      $     C     ONE HOP JAMMED
0056      $     C     ONE HOP JAMMED
0057      3200  IF (REG1) THEN
           XCOM=YK+2.0*D*RHON+RHOT
           CALL ADQUA2(0.0D,Y,ANSWER,DGX,F31A,1.0-9,WORK1,STACK1)
0058      $     C     ONE HOP JAMMED
0059      $     C     ONE HOP JAMMED
0060      $     C     ONE HOP JAMMED
0061      $     C     ONE HOP JAMMED
0062      $     C     ONE HOP JAMMED
0063      $     C     ONE HOP JAMMED
0064      $     C     ONE HOP JAMMED
0065      $     C     ONE HOP JAMMED
0066      $     C     ONE HOP JAMMED
0067      $     C     ONE HOP JAMMED
0068      $     C     ONE HOP JAMMED
0069      $     C     ONE HOP JAMMED
0070      $     C     ONE HOP JAMMED
0071      $     C     ONE HOP JAMMED
0072      $     C     ONE HOP JAMMED
0073      $     C     ONE HOP JAMMED
0074      $     C     ONE HOP JAMMED
0075      $     C     ONE HOP JAMMED
0076      $     C     ONE HOP JAMMED
0077      $     C     ONE HOP JAMMED
0078      $     C     ONE HOP JAMMED

```

```

PDF-11 FORTRAN-77 V4.0-1      10:45:36   16-Ju1-86      Page 40
CLPL35GR.FTM;6    /F77/MR

0163      CALL TEST2(K00,3400)
0164      PAR=PART+5.0*ANSWER/(BIGK*BIGK)
0165      CALL ADQUA2(TAU,Y,ANSWER,DX,X33A2,1,D-9,WORK1,STACK1,
0166          HEAP1,30,K00)
0167      CALL TEST2(K00,3401)
0168      PZ1=PART+ANSWER/DX1(BIGK,3)
0169      BARG1=D$QRT(4.0*D$RHT*YT2)
0170      I01=0
0171      CALL DXBT1(3402)
0172      PART=3.D0*Q1*Q1*D$EXP(BARG1-YT2-RHOT)*B1/BIGK
0173      XCOM=YTK+2.00*RHOT
0174      CALL ADQUA2(Y-TAU,TAU2,ANSWER,DX,F33B1,1,D-9,WORK1,STACK1,
0175          HEAP1,30,K00)
0176      CALL TEST2(K00,3402)
0177      PART=PART+3.00*Q1*ANSWER/(BIGK*BIGK)
0178      XCOM=YK+3.D0*RHOT
0179      CALL ADQUA2(Y-TAU,TAU2,ANSWER,DX,F33B2,1,D-9,WORK1,STACK1,
0180          HEAP1,30,K00)
0181      CALL TEST2(K00,3403)
0182      PZ1=PART+ANSWER/DX1(BIGK,3)
0183      ELSE
0184          PZ1=0.00
0185      END IF
0186      CONTINUE
0187      RETURN
0188 9000

```

P00-11 FORTRAN-77 v4.0-1 /F77/NR
CLPL3SGR.FTN;6 Page 41

PDP-11 FORTRAN-77 V4.0-1 10:46:11 16-Jul-86 Page 43
 CLIP13SGR.FTN;6 /F77/MR

0001 SUBROUTINE BPROD(IDENT)

C COMPUTE TWO BESSSEL FUNCTIONS, ARGUMENTS AND RESULTS IN COMMON

C

```

0002 IMPLICIT DOUBLE PRECISION(A-H,0-2)
0003 COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
0004 CALL DXB(IDENT)
0005 CALL DIRET(BARG2,102,B2,KODE)
0006 CALL TEST(IDENT+1)
0007 RETURN
0008 END

```

PDP-11 FORTRAN-77 V4.0-1 10:46:12 16-Jul-86 Page 44
 CLIP13SGR.FTN;6 /F77/MR

0001 DOUBLE PRECISION FUNCTION F30A1(U)

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJAM=0,

C FIRST INTEGRAL

C

```

0002 IMPLICIT DOUBLE PRECISION(A-H,0-2)
0003 COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
0004 COMMON /DENPAR/ B1GK, AAB, BAB, LJAM, AAB2, BAB2, ABBAB,
0005      $ TAU, TAU2, TAU3, TAUK2, TAUK3
0006 COMMON /PARDEN/ RHOM, RHOT
0007 COMMON /QUES/ Q0, Q1
0008 COMMON /XCOM/ XCON
0009 COMMON /OUTER/ XXX, XXX
0010 COMMON /INNER/ WWW, WWWW
0011 LJAM=U
0012 101=0
0013 CALL ADQUA3(U-TAU,TAU,ANSWER,DGTEN,F30ABA,1,D-10,
0014      $ WORK,STACK,HEAP,30,K00)
0015 BARG1=DSQRT(4.00*RHOM*(WWW-U))
0016 101=0
0017 CALL DXBT(3112)
0018 F30A2=DEXP(BARG1-XCON)*B1*ANSWER
0019 RETURN
0020 END

```

J-22 PDP-11 FORTRAN-77 V4.0-1 10:46:17 16-Jul-86 Page 45
 CLIP13SGR.FTN;6 /F77/MR

0001 DOUBLE PRECISION FUNCTION F30ABA(U)

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJAM=0,

C SECOND INTEGRAL S INNER INTEGRAL; ALSO L=3, LJAM=1, SECOND

C REGION, THIRD INTEGRAL, INNER INTEGRAL.

C

```

0002 IMPLICIT DOUBLE PRECISION(A-H,0-2)
0003 COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
0004 COMMON /DENPAR/ B1GK, AAB, BAB, LJAM, AAB2, BAB2, ABBAB,
0005      $ TAU, TAU2, TAU3, TAUK2, TAUK3
0006 COMMON /PARDEN/ RHOM, RHOT
0007 COMMON /QUES/ Q0, Q1
0008 COMMON /XCOM/ XCON
0009 COMMON /OUTER/ XXX, XXX
0010 BARG1=DSQRT(4.00*RHOM*(XXX-U))
0011 BARG2=DSQRT(4.00*RHOM*(WWW-U))
0012 101=1
0013 102=0
0014 CALL BPROD(3110)
0015 S=DSQRT(2.00*U/RHOM)
0016 F30A1=S*((DEXP(BARG1)+BARG2-XCON)*B1)*B2
0017 RETURN
0018 END

```

PDP-11 FORTRAN-77 V4.0-1 10:46:14 16-Jul-86 Page 45
 CLIP13SGR.FTN;6 /F77/MR

0001 SUBROUTINE DXBT(1D)

C CALL DBESI AND TEST RETURN CODE

C

```

0002 IMPLICIT DOUBLE PRECISION(A-H,0-2)
0003 COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
0004 CALL DIRET(BARG1,101,B1,KODE)
0005 CALL TEST(1D)
0006 RETURN
0007 END

```

PDP-11 FORTRAN-77 V4.0-1 10:46:15 16-Jul-86 Page 46
 CLIP13SGR.FTN;6 /F77/MR

0001 DOUBLE PRECISION FUNCTION F30A2(U)

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJAM=0,

C SECOND INTEGRAL

C

```

0002 IMPLICIT DOUBLE PRECISION(A-H,0-2)
0003 EXTERNAL DGTEN, F30ABA
0004 COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
0005 COMMON /INWORK/ WORK(30), STACK(30)
0006 COMMON /DENPAR/ B1GK, AAB, BAB, LJAM, AAB2, BAB2, ABBAB,
0007      $ TAU, TAU2, TAU3, TAUK2, TAUK3
0008 COMMON /PARDEN/ RHOM, RHOT
0009 COMMON /QUES/ Q0, Q1
0010 COMMON /XCOM/ XCON
0011 COMMON /OUTER/ XXX, XXX
0012 COMMON /INNER/ WWW, WWWW
0013 LJAM=U
0014 CALL TEST2(K3D,3112)
0015 BARG1=DSQRT(4.00*RHOM*(XXX-U))
0016 101=0
0017 CALL DXBT(3112)
0018 F30A2=DEXP(BARG1-XCON)*B1*ANSWER
0019 RETURN
0020 END

```

PDP-11 FORTRAN-77 V4.0-1 10:46:17 16-Jul-86 Page 47
 CLIP13SGR.FTN;6 /F77/MR

0001 DOUBLE PRECISION FUNCTION F30ABA(U)

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJAM=0,

C SECOND INTEGRAL S INNER INTEGRAL; ALSO L=3, LJAM=1, SECOND

C REGION, THIRD INTEGRAL, INNER INTEGRAL.

C

```

0002 IMPLICIT DOUBLE PRECISION(A-H,0-2)
0003 COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
0004 COMMON /DENPAR/ B1GK, AAB, BAB, LJAM, AAB2, BAB2, ABBAB,
0005      $ TAU, TAU2, TAU3, TAUK2, TAUK3
0006 COMMON /PARDEN/ RHOM, RHOT
0007 COMMON /QUES/ Q0, Q1
0008 COMMON /XCOM/ XCON
0009 COMMON /OUTER/ XXX, XXX
0010 COMMON /INNER/ WWW, WWWW
0011 BARG1=DSQRT(4.00*RHOM*(WWW-U))
0012 CALL DBESI(BARG1,0,B1,KODE)
0013 BARG2=DSQRT(4.00*RHOM*(WWW-U))
0014 CALL DBESI(BARG2,0,B2,KODE)
0015 CALL TEST(1311)
0016 F30ABA=B1*B2
0017 RETURN
0018 END

```

PDP-11 FORTRAN-77 V4.0-1 10:46:20 16-Jul-86 Page 48

CLPL3SGR.FTN;6 /F77/WR

0001 DOUBLE PRECISION FUNCTION F30B1(U)

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJAM=0,
C THIRD REGION, FIRST INTEGRAL

C

```

0002 IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0003 COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
0004      $ COMMON /DENPAR/ BIGK, AAB, BAB, LJAM, AAB2, BAB2, ABBAB,
0005      $ COMMON /PARDEN/ RHON, RHOT
0006      $ COMMON /QUES/ Q0, Q1
0007      $ COMMON /XCOM/ XCOM
0008      $ COMMON /OUTER/ XXX, XXXK
0009      $ BARG1=DSORT(4.D0*RHOM*(XXX-U))
0010      $ BARG2=DSORT(4.D0*RHOM*(U-TAU))
0011      $ 101=0
0012      $ 102=0
0013      CALL BPROD(3120)
0014      F30B1=DEXP(BARG1+BARG2-XCOM)*B1*B2
0015      RETURN
0016      END

```

J-23

PDP-11 FORTRAN-77 V4.0-1 10:46:21 16-Jul-86 Page 49

CLPL3SGR.FTN;6 /F77/WR

0001 DOUBLE PRECISION FUNCTION F30B2(U)

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJAM=0,
C SECOND REGION, SECOND INTEGRAL

C

```

0002 IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0003 EXTERNAL DGTEN, F30ABA
0004 COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
0005 COMMON /INWORK/ WORK(30), STACK(30), HEAP(30)
0006      $ COMMON /DENPAR/ BIGK, AAB, BAB, LJAM, AAB2, BAB2, ABBAB,
0007      $ COMMON /PARDEN/ RHON, RHOT
0008      $ COMMON /QUES/ Q0, Q1
0009      $ COMMON /XCOM/ XCOM
0010      $ COMMON /OUTER/ XXX, XXXK
0011      $ COMMON /INNER/ WHH, WHK
0012      $ WHH=U
0013      CALL ADQUA3(U-TAU, TAU, ANSWER, DGTEN, F30ABA, 1.D-10,
0014      $ WORK, STACK, HEAP, 30, KOD)
0015      $ CALL TEST2(KOD, 3120)
0016      $ BARG1=DSORT(4.D0*RHOM*(XXX-U))
0017      $ 101=0
0018      $ CALL QXBT(3121)
0019      $ F30B2=DEXP(BARG1-XCOM)*B1*ANSWER
0020      $ RETURN
      END

```

PDP-11 FORTRAN-77 V4.0-1 10:46:20 16-Jul-86 Page 50

CLPL3SGR.FTN;6 /F77/WR

0001 DOUBLE PRECISION FUNCTION F31AU()

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJAM=1,
C FIRST REGION

C

```

0002 IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0003 COMMON /DENPAR/ BIGK, AAB, BAB, LJAM, AAB2, BAB2, ABBAB,
0004      $ COMMON /PARDEN/ TAU, TAU2, TAU3, TAUK2, TAUK3
0005 COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
0006 COMMON /QUES/ Q0, Q1
0007 COMMON /XCOM/ XCOM
0008 COMMON /OUTER/ XXX, XXXK
0009      $ BARG1=DSORT(4.D0*RHOT*(XXXK-U/BIGK))
0010      $ BARG2=DSORT(2.D0*RHOM*U)
0011      $ 101=0
0012      $ 102=1
0013      CALL BPROD(3210)
0014      F31A=DEXP(BARG1+BARG2-XCOM)*B1*B2*DSORT(2.D0*U/RHOM)
0015      RETURN
0016      END

```

PDP-11 FORTRAN-77 V4.0-1 10:46:25 16-Jul-86 Page 51

CLPL3SGR.FTN;6 /F77/WR

0001 DOUBLE PRECISION FUNCTION F31B1(U)

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJAM=1,
C SECOND REGION, FIRST INTEGRAL

C

```

0002 IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0003 COMMON /DENPAR/ BIGK, AAB, BAB, LJAM, AAB2, BAB2, ABBAB,
0004      $ COMMON /PARDEN/ TAU, TAU2, TAU3, TAUK, TAUK2, TAUK3
0005 COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
0006 COMMON /QUES/ Q0, Q1
0007 COMMON /XCOM/ XCOM
0008 COMMON /OUTER/ XXX, XXXK
0009      $ BARG1=DSORT(4.D0*RHOT*(XXXK-U/BIGK))
0010      $ UK=U*BIGK
0011      $ BARG2=DSORT(4.D0*RHOT*(XXXK-UK))
0012      $ 101=1
0013      $ 102=0
0014      CALL BPROD(3220)
0015      $ 5=DSORT(12.D0/U/RHOM)
0016      $ F31B1=DEXP(BARG1+BARG2-XCOM-U*UK)*B1*B2*S
0017      $ RETURN
0018      END

```

J-24 PDP-11 FORTRAN-77 V4.0-1 10:46:27 16-Jul-86 Page 52 CLPL3SGR.FTN:6 /F77/MR

0001 C DOUBLE PRECISION FUNCTION F31B2(U)

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJMM=1,

C SECOND REGION, SECOND INTEGRAL

C

```

0002 IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003 COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
0004      $ COMMON /DENPAR/ BIGK, AAB, BAA, LJMM, AAB2, BAB2, ABBAB,
0005      $ COMMON /PARDEN/ RHON, RHOT
0006      COMMON /QUES/ QO, Q1
0007      COMMON /XCON/ XCON
0008      COMMON /OUTER/ XXX, XXX
0009      BARG1=DSORT(4.0*DPRHOM*U)
0010      UK=U/BIGK
0011      BARG2=DSORT(4.0*DPRHOT*(XXXK-UK))
0012      101=0
0013      102=0
0014      CALL BPROD(3222)
0015      F31B2=DEXP(BARG1+BARG2-XCON-U+UK)*B1*B2
0016      RETURN
0017      END

```

PDP-11 FORTRAN-77 V4.0-1 10:46:29 16-Jul-86 Page 53 CLPL3SGR.FTN:6 /F77/MR

0001 C DOUBLE PRECISION FUNCTION F31B3(U)

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJMM=1,

C SECOND REGION, THIRD INTEGRAL

C

```

0002 IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003 EXTERNAL DETEN, F30ABA
0004 COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
0005 COMMON /WORK/ WORK(30), STACK(30), HEAP(30)
0006      $ COMMON /DENPAR/ BIGK, AAB, BAA, LJMM, AAB2, BAB2, ABBAB,
0007      COMMON /PARDEN/ RHON, RHOT
0008      COMMON /QUES/ QO, Q1
0009      COMMON /XCON/ XCON
0010      COMMON /OUTER/ XXX, XXX
0011      COMMON /INNER/ WMN, WMNK
0012      WMN=U
0013      CALL ADQUA3(U-TAU, TAU, ANSWER, DGTE, F30ABA, 1, D-10,
0014      WORK, STACK, HEAP, 30, KOD)
0015      $ CALL TEST2(KOD, 3222)
0016      BARG1=DSORT(4.0*DPRHOT*(XXXK-UK))
0017      UK=U/BIGK
0018      CALL DXBT(3222)
0019      F31B3=DEXP(BARG1-XCON-U+UK)*B1*ANSWER
0020      RETURN

```

PDP-11 FORTRAN-77 V4.0-1 10:46:31 16-Jul-86 Page 54 CLPL3SGR.FTN:6 /F77/MR

0001 C DOUBLE PRECISION FUNCTION F31C2(U)

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJMM=1,

C THIRD REGION, SECOND INTEGRAL

C

```

0002 IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003 COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
0004      $ COMMON /DENPAR/ BIGK, AAB, BAA, LJMM, AAB2, BAB2, ABBAB,
0005      $ COMMON /PARDEN/ RHON, RHOT
0006      COMMON /QUES/ QO, Q1
0007      COMMON /XCON/ XCON
0008      COMMON /OUTER/ XXX, XXX
0009      UK=U/BIGK
0010      BARG1=DSORT(4.0*DPRHOT*(XXXK-UK))
0011      BARG2=DSORT(4.0*DPRHOT*(U-TAU))
0012      101=0
0013      102=0
0014      CALL BPROD(3330)
0015      F31C2=DEXP(BARG1+BARG2-XCON-U+UK)*B1*B2
0016      RETURN
0017      END

```

PDP-11 FORTRAN-77 V4.0-1 10:46:33 16-Jul-86 Page 55 CLPL3SGR.FTN:6 /F77/MR

0001 C DOUBLE PRECISION FUNCTION F33A1(U)

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJMM=3,

C FIRST INTEGRAL

C

```

0002 IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003 COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
0004      $ COMMON /DENPAR/ BIGK, AAB, BAA, LJMM, AAB2, BAB2, ABBAB,
0005      $ COMMON /PARDEN/ RHON, RHOT
0006      COMMON /QUES/ QO, Q1
0007      COMMON /XCON/ XCON
0008      COMMON /OUTER/ XXX, XXX
0009      UK=U/BIGK
0010      BARG1=DSORT(8.0*DPRHOT*UK)
0011      BARG2=DSORT(4.0*DPRHOT*(XXXK-UK))
0012      101=1
0013      102=0
0014      CALL BPROD(3410)
0015      S=DSORT(2.0*DPRHOT)
0016      F33A1=S*( (DEXP(BARG1+BARG2-XCON)*B1)*B2 )
0017      RETURN
0018      END

```

```

PDP-11 FORTRAN-77 V4.0-1 10:46:35 16-Jul-86 Page 56
CL1PL3SGR.FTN;6 /F77/NR

0001 C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJAM=3,
C SECOND INTEGRAL.
      IMPLICIT DOUBLE PRECISION(A-H,0-Z)
      EXTERNAL DGTEEN, F33ABA
      COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
      COMMON /INWORK/ WORK(30), STACK(30), HEAP(30)
      COMMON /DEMPAR/ BIGK, AAB, BAB, LJAM, AAB2, BAB2, ABBAB,
      $      TAU, TAU2, TAU3, TAUK, TAUK2, TAUK3
      COMMON /PARDEN/ RHOM, RHOI
      COMMON /QUES/ Q0, Q1
      COMMON /XCON/ XCON
      COMMON /OUTER/ XXX, XXXX
      COMMON /INNER/ MMN, MMK
      UK=U/BIGK
      BARG1=DSQRT(4.0D0*RHOI*(XXXK-UK))
      CALL ADQUA3(U-TAU, TAU, ANSWER, DGTEEN, F33ABA, 1, D-10,
      WORK, STACK, HEAP, 30, KOD)
      $      CALL TEST2(KOD, 3412)
      BARG1=DSQRT(4.0D0*RHOI*(XXXK-UK))
      101=0
      CALL DBIT(3112)
      F33A2=EXP(BARG1-XCON)*B1*ANSWER
      RETURN
      END
      -0020

```

```

PDP-11 FORTRAN-77 V4.0-1 10:46:35 16-Jul-86 Page 56
CL1PL3SGR.FTN;6 /F77/NR

0001 C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJAM=3,
C THIRD REGION, FIRST INTEGRAL.
      IMPLICIT DOUBLE PRECISION(A-H,0-Z)
      COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
      COMMON /INWORK/ BIGK, AAB, BAB, LJAM, AAB2, BAB2, ABBAB,
      $      TAU, TAU2, TAU3, TAUK, TAUK2, TAUK3
      COMMON /PARDEN/ RHOM, RHOI
      COMMON /QUES/ Q0, Q1
      COMMON /XCON/ XCON
      COMMON /OUTER/ XXX, XXXX
      COMMON /INNER/ MMN, MMK
      UK=U/BIGK
      BARG1=DSQRT(4.0D0*RHOI*(XXXK-UK))
      CALL ADQUA3(U-TAU, TAU, ANSWER, DGTEEN, F33ABA, 1, D-10,
      WORK, STACK, HEAP, 30, KOD)
      $      CALL TEST2(KOD, 3420)
      F33B1=EXP((BARG1+BARG2-XCON)*B1*B2
      RETURN
      END

PDP-11 FORTRAN-77 V4.0-1 10:46:36 16-Jul-86 Page 59
CL1PL3SGR.FTN;6 /F77/NR

0001 C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJAM=3,
C SECOND INTEGRAL, SECOND INTEGRAL.
      IMPLICIT DOUBLE PRECISION(A-H,0-Z)
      EXTERNAL DGTEEN, F33ABA
      COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
      COMMON /INWORK/ WORK(30), STACK(30), HEAP(30)
      COMMON /DEMPAR/ BIGK, AAB, BAB, LJAM, AAB2, BAB2, ABBAB,
      $      TAU, TAU2, TAU3, TAUK, TAUK2, TAUK3
      COMMON /PARDEN/ RHOM, RHOI
      COMMON /QUES/ Q0, Q1
      COMMON /XCON/ XCON
      COMMON /OUTER/ XXX, XXXX
      COMMON /INNER/ MMN, MMK
      UK=U/BIGK
      MMK=UK
      CALL ADQUA3(U-TAU, TAU, ANSWER, DGTEEN, F33ABA, 1, D-10,
      WORK, STACK, HEAP, 30, KOD)
      $      CALL TEST2(KOD, 3420)
      BARG1=DSQRT(4.0D0*RHOI*(XXXK-UK))
      101=0
      CALL DXBT(3421)
      F33B2=EXP((BARG1-XCON)*B1*ANSWER
      RETURN
      END

```

POP-11 FORTRAN-77 V4.0-1 10:46:44 16-Jul-86 Page 60
 CLPL3SGR.FTN;6 /F77/MR

0001 DOUBLE PRECISION FUNCTION F32A(U)

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJAH=2,
 C FIRST REGION

```

    C
    0002  IMPLICIT DOUBLE PRECISION(A-H,0-2)
    0003  COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
    0004  S COMMON /DEMPAR/ BIGK, AAB, BAB, LJAM, AAB2, BAB2, ABBAB,
    0005          S TAU, TAU2, TAU3, TAUK, TAUK2, TAUK3
    0006  COMMON /PARDEN/ RHON, RHOI
    0007  COMMON /QUES/ QO, Q1
    0008  COMMON /XCON/ XCON
    0009  UK=U/BIGK
    0010  BARG1=DSQRT(4.D0*RHON*(XXX-U))
    0011  BARG2=DSQRT(2.D0*RHOI*UK)
    0012  101=0
    0013  102=1
    0014  CALL BPROD(3320)
    0015  F32A=DEXP(BARG1+BARG2-XCON)*B1*B2*DSQRT(2.D0*U/RHON)
    0016  RETURN
    0017  END
  
```

C

-1 POP-11 FORTRAN-77 V4.0-1 10:46:46 16-Jul-86 Page 61
 CLPL3SGR.FTN;6 /F77/MR

0001 DOUBLE PRECISION FUNCTION F32B1(U)

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJAH=2,
 C SECOND REGION, FIRST INTEGRAL

```

    C
    0002  IMPLICIT DOUBLE PRECISION(A-H,0-2)
    0003  COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
    0004  S COMMON /DEMPAR/ BIGK, AAB, BAB, LJAM, AAB2, BAB2, ABBAB,
    0005          S TAU, TAU2, TAU3, TAUK, TAUK2, TAUK3
    0006  COMMON /PARDEN/ RHON, RHOI
    0007  COMMON /QUES/ QO, Q1
    0008  COMMON /XCON/ XCON
    0009  UK=U/BIGK
    0010  BARG1=DSQRT(8.D0*RHOI*UK)
    0011  BARG2=DSQRT(4.D0*RHON*(XXX-U))
    0012  101=1
    0013  102=0
    0014  CALL BPROD(3320)
    0015  S=DSQRT(2.D0*U/RHON)
    0016  F32B1=DEXP(BARG1+BARG2-XCON-UK+U)*B1*B2*S
    0017  RETURN
    0018
  
```

POP-11 FORTRAN-77 V4.0-1 10:46:48 16-Jul-86 Page 62
 CLPL3SGR.FTN;6 /F77/MR

0001 DOUBLE PRECISION FUNCTION F32B2(U)

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJAH=2,
 C SECOND REGION, SECOND INTEGRAL

```

    C
    0002  IMPLICIT DOUBLE PRECISION(A-H,0-2)
    0003  COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, 101, 102
    0004  S COMMON /DEMPAR/ BIGK, AAB, BAB, LJAM, AAB2, BAB2, ABBAB,
    0005          S TAU, TAU2, TAU3, TAUK, TAUK2, TAUK3
    0006  COMMON /PARDEN/ RHON, RHOI
    0007  COMMON /QUES/ QO, Q1
    0008  COMMON /XCON/ XCON
    0009  UK=U/BIGK
    0010  BARG1=DSQRT(4.D0*RHOI*UK)
    0011  101=0
    0012  BARG2=DSQRT(4.D0*RHON*(XXX-TAU-U))
    0013  102=0
    0014  CALL BPROD(3322)
    0015  F32B2=DEXP(BARG1+BARG2-XCON-UK+U)*B1*B2
    0016  RETURN
    0017  END
  
```

C

-1 POP-11 FORTRAN-77 V4.0-1 10:46:50 16-Jul-86 Page 63
 CLPL3SGR.FTN;6 /F77/MR

0001 DOUBLE PRECISION FUNCTION F32B3(U)

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJAH=2,
 C SECOND REGION, THIRD INTEGRAL

```

    C
    0002  IMPLICIT DOUBLE PRECISION(A-H,0-2)
    0003  EXTERNAL DSEN, F33ABA
    0004  COMMON /INWORK/ WORK(30), STACK(30), HEAP(30)
    0005  COMMON /DEMPAR/ BIGK, AAB, BAB, LJAM, AAB2, BAB2, ABBAB,
    0006  COMMON /PARDEN/ RHON, RHOI
    0007  COMMON /QUES/ QO, Q1
    0008  COMMON /XCON/ XCON
    0009  COMMON /OUTER/ XXX, XXX
    0010  COMMON /INNER/ WWW, WWW
    0011  UK=U/BIGK
    0012  WWW=UK
    0013  CALL ADQUA(U-TAU, TAU, ANSWER, DGETN, F33ABA, 1.0-10,
    0014          WORK, STACK, HEAP, 30, KOD)
    0015  S=CALL TEST2(KOD, 3322)
    0016  BARG1=DSQRT(4.D0*RHON*(XXX-U))
    0017  101=0
    0018  CALL DX8T(3324)
    0019  F32B3=DEXP(BARG1-XCON-UK+U)*B1*ANSWER
    0020  RETURN
    0021
  
```

```

PDP-11 FORTRAN-77 V4.0-1      10-46:52   16-Jul-86   Page 64
CLPL3SGR.FTN;6    /F77/NR

0001      DOUBLE PRECISION FUNCTION F32C2(U)

C INTEGRAND FUNCTION FOR SIGNAL CHANNEL DENSITY, L=3, LJM=2,
C THIRD REGION, SECOND INTEGRAL

C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
COMMON /LOCALS/ BARG1, BARG2, B1, B2, KODE, T01, T02, 102
COMMON /DENPAR/ BICK, BAB, BAB, LJM, AAB2, BAB2, AABAB
COMMON /TAU/ TAU1, TAU2, TAU3, TALK, TALK2, TALK3
COMMON /PARDEN/ RHOH, RHOI
COMMON /ONES/ Q0, Q1
COMMON /XCOM/ XCOM
COMMON /OUTER/ XXX, XXX
UK=U/BIGK
BARG1=DSORT(4.00*RHOH*(XXX-U))
T01=0
BARG2=DSORT(4.00*RHOH*(UK-TALK))
T02=0
CALL BPROD(3330)
F32C2=0EXP(BARG1+8*BARG2-XCOM-UK+U)*B1*B2
RETURN
END

```

```

PPF-11 FORTRAN-77 V4.0-1 /F77/LR
CL1PL35GR.FTN:6 Page 65 10:46:54 16-Jul-86

0001      SUBROUTINE TIES(JSUB,MM,PTIE)
0002      C COMPUTE PROBABILITY OF CORRECT DECISION GIVEN THAT
0003      C SEVERAL SATURATED CHANNELS ARE TIED
0004      C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0005      DIMENSION JSUB(MM), LL0W(7), LINC(7), LUP(7), MU(7),
0006      P2LM(2-8),
0007      S LOGICAL*1 GO
0008      COMMON /DENPAR/ B1SK, AAB, BAB, LJM, AAB2, BAB2, ABBAB,
0009      S COMMON /PARDEN/ TAU, TAU3, TAUK, TAUK2, TAUK3
0010      COMMON /QUES/ QO, QI
0011      C NUMBER OF NON-SIGNAL CHANNELS
0012      MMU=MM-1
0013      PTIE=0.00
0014      CUE0=DEAR(-TAU)
0015      CUE1=DXP(-TAUK)
0016      P1L=DX1(00,-3-JSUB(1))*DX1(01,JSUB(1))
0017      DO 10 I=2,MM
0018      P2LM(1)=DX1(CUE0,-3-JSUB(1))*DX1(CUE1,JSUB(1))
0019      CONTINUE
0020      10 CONTINUE
0021      C SET UP VECTOR LOOP PARAMETERS
0022      C DO 20 I=1,MM-1
0023      LL0W(I)=0
0024      LINC(I)=1
0025      LUP(I)=1
0026      CONTINUE
0027      PTIE=0.00
0028      C START LOOP ON THE TIE EVENTS
0029      CALL VLINIT(MU,LL0W,MM-1)
0030      MUSUM=0
0031      DO 40 I=1,MM-1
0032      MUSUM=MUSUM+MU(I)
0033      CONTINUE
0034      FRAC=1.0/(1.00+MUSUM)
0035      PROD=PROD*(1.00-P2LM(M))
0036      IF(M>2,MM)
0037      THEN
0038      PROD=PROD*P2LM(M)
0039      ELSE
0040      PROD=PROD*(1.00-P2LM(M))
0041      END IF
0042      CONTINUE
0043      PTIE=FRAC*PROD*PTIE
0044      CALL VLITER(MU,LL0W,LUP,LINC,MM-1,GO)
0045      IF(GO) GOTO 30
0046      PTIE=PTIE*P1L
0047      RETURN
0048      END

```

```

0001      DOUBLE PRECISION FUNCTION PUNJAM(ETA)
          C FUNCTION FOR UNJAMMED P(E) FOR OPT. THRESHOLD SEARCH
          C
          C NOTE: WHEN JAMMING EVENT IS (0.0,...,0), THE VARIABLES
          C       BIGK, AAB, BAB, TAUK, TAUK2, AND TAUK3 ARE NOT
          C       USED IN THE COMPUTATIONS, AND HENCE DO NOT NEED
          C       TO BE SET UP BEFORE CALLING PSEL FROM THIS FUNCTION
          C
          C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          DIMENSION NOJAM(4)
          COMMON /INPUTS/ DEBTRL(3),NSLOTS,K,MN
          COMMON /DENPAR/ BIGK, AAB, BAB, LJAM, AAB2, BAB2, ABBAB,
          TAU, TAU2, TAU3, TAUK, TAUK2, TAUK3
          DATA NOJAM/0,0,0,0/
          LJAM=0
          TAU=ETA
          TAU2=TAU+ETA
          TAU3=TAU2+ETA
          CALL PSEL(NOJAM,MN,P)
          PUNJAM=P
          RETURN
          END

```

```

0001      SUBROUTINE SETTAU(MN,PE00)
          C SEARCH FOR OPTIMUM THRESHOLD IN ABSENCE OF JAMMING
          C
          C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          EXTERNAL PUNJAM
          COMMON /DENPAR/ BIGK, AAB, BAB, LJAM, AAB2, BAB2, ABBAB,
          TAU, TAU2, TAU3, TAUK, TAUK2, TAUK3
          $   COMMON /PARDEN/ RHOM, RHOI
          COMMON /QUEST/ QD, QI
          LJAM=0
          C GUESS BASED ON PREVIOUS RESULTS
          C
          GUESS=8.00
          STEP=1.00
          C FOR MN=4, L=3 THE OPTIMUM THRESHOLD IS 8.15
          CALL MINSER(PUNJAM,PENIN,TAUOPT,STEP,GUESS,0.00,
          0009          50.00,0.01DD)
          0010          $   TAU=TAUOPT
          TAU2=TAU+TAU
          TAU3=TAU2+TAU
          PE00=PENIN
          RETURN
          END

```

J. S. LEE ASSOCIATES, INC.

APPENDIX K COMPUTER PROGRAM FOR SELF-NORMALIZING RECEIVER WITH M=2, L=2

The following pages contain the source code listing for the FORTRAN-77 program used for numerical computations of the performance of the self-normalizing receiver when $M=2$ and $L=2$, with jamming fraction γ as a parameter.

PDP-11 FORTRAN-77 V4.0-1 09:50:01 17-Jul-86 Page 1
 SELF2M2.FTN:13 /F77/TR:BLOCKS/MR

0001 PROGRAM SELF22

 C SELF NORMALIZING RECEIVER, M=2, L=2

 C ANALYSIS: L.E. MILLER
 C PROGRAM: R.H. FRENCH

0002 IMPLICIT DOUBLE PRECISION (A,H,O-2)

 CHARACTER(13) FILENAME
 REAL DEBNJ(126), PEL06(126)
 COMMON /ROSE/ RHOM, RHOT, GAMMA, BIGK
 COMMON /PARMS/ MO, MJ, MG, DEBNOL(5), GAMLST(10),
 DEBNAL(126), DJ, DBJ0,
 CALL ERSET(29, .TRUE... .FALSE... .TRUE.. ,15)

0003 DO 900 I=1,M0
 EBNO=EBNO/2.00
 IONUT=DEBNOL(10)

0004 DO 800 I=1,NG
 GAMMA=GAMLST(1IG)
 I0=GAMMA*MSLOTS+0.500
 IOUT=1000.DP*GAMMA+0.500

0005 C PROGRESS FILE

0006 WRITE(FILENAME,1) 100IUT,IGOUT
 1 FORMAT('S22',12.2,I4.1, .DAT')

0007 WRITE(FILENAME,2) FILENAME
 2 FORMAT(' WORKING ON FILE ',A13)

0008 OPEN(UNIT=4,FILE=FILENAME,STATUS='OLD',ERR=810,
 FORM='UNFORMATTED',ACCESS='SEQUENTIAL',

0009 C HAVE A PROGRESS FILE, READ IT

0010 READ(4) EBNOIN, GAMMIN, DBJ0IN, DJIN
 IF(EBNOIN.NE.DBJ0IN) OR. GAMMIN.NE.GAMMA .OR.
 DBJ0IN.NE.DBJ0 .OR. DJIN.NE.DJ) STOP 'FILE CORRUPT.'

0011 JJ=0
 801 JJ=JJ+1
 READ(4,END=802) DEBNJ(JJ), PEL06(JJ)

0012 GOTO 801
 802 CLOSE(UNIT=4)

0013 GOTO 820

0014 C NO FILE, MUST CREATE IT

0015 810 OPEN(UNIT=4,FILE=FILENAME,STATUS='NEW',FORM='UNFORMATTED',
 ACCESS='SEQUENTIAL')
 811 WRITE(4) DEBNOL(10), GAMMA, DBJ0, DJ
 CLOSE(UNIT=4)

0034 .JJ=1
 C KEEP ON GOING

0035 C 820 DO 700 IJ=JJ,MJ
 0036 WRITE(15,821) IJ
 0037 FORMAT(1, IJ = ,13)
 DEBNJ(IJ)=DBJ0*(IJ-1)*DJ
 DEBNJ(IJ)=DBEBRJ
 EBKJ=10.0D+*(DBEBRJ/10.00)
 RHOKJ=8M/72.00
 RHOKJ=64.00*RHOKJ
 PHOT=RHOKJ*ARHOM/(RHOKJ+RHOM)
 BIGK=RHOKJ*RHOT
 CALL PSUBE(IQ,PF)
 PEL06(IJ)=DLOG10(PF)
 OPEN(UNIT=4,FILE=FILENAME,STATUS='OLD',ACCESS='APPEND',
 FORM='UNFORMATTED')
 WRJTE(IJ) DEBNJ(IJ), PEL06(IJ)
 CLOSE(UNIT=4)

0041 0042 700 CONTINUE
 OPEN(UNIT=4,FILE=FILENAME,STATUS='NEW',ACCESS='SEQUENTIAL',
 FORM='UNFORMATTED')
 WRJTE(4) H, L, DEBNAL(10), GAMMA, MSLOTS, MJ, DEBNJ, PEL06
 CLOSE(UNIT=4)

0043 0044 0045 0046 0047 0048 0049 0050 0051 0052 0053 0054 0055 0056 0057 0058 0059
 WRJTE(6,710) M, L, DEBNOL(10), GAMMA, MSLOTS,
 (DEBNJ(IJ),10.DP*PEL06(IJ),IJ=1,MJ)
 FORMAT('1SELF-NORMALIZING RECEIVER FOR M=11, AND L=11/
 5X, EB/MO = ',F8.5,' dB ; 5X, 'GAMMA= ',1PD10.3,5X, 'MSLOTS= ',14/
 EB/NJ (dB) ,8X, P(E))/END(4X,OPFA.1.8X,JP10.3/))
 CONTINUE
 CONTINUE
 STOP 'PLEASE PURGE S22*.DAT'
 END

PDP-11 FORTRAN-77 V4.0-1 09:50:10 17-Jul-86

SELF12M2.FTN:13 /F77/TR:BLOCKS/MR

Page 3 PDP-11 FORTRAN-77 V4.0-1 09:50:10 17-Jul-86

Page 4

```
0001      SUBROUTINE GET
C      INTERACTIVE RUN PARAMETER INPUTS
C
0002      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003      DIMENSION DEAM(10), DOLST(5)
0004      CHARACTER*8 REPLY, BLANKS
0005      COMMON /PARMS/ NO, NJ, NG, DEBNOL(5), GAMLST(10),
DEBNJ(126), DJ, DBJ0
0006      DATA DEAM/1.D-3, 2.D-3, 5.D-3,
1.D-2, 2.D-2, 5.D-2,
1.D-1, 2.D-1, 5.D-1, 1.00/
0007      DATA DOLST/ 13.3524700, 12.313300, 10.9444300, 14.3925300,
16.02713500/
0008      DATA BLANKS/, /.
0009      WRITE(5,2)
0010      2 FORMAT(1H HOW MANY EB/NJ? [1] :, $)
0011      READ(5,3,ERR=1) NO
0012      3 FORMAT([1])
0013      IF(NO.EQ.0) NO=1
0014      IF(NO.LT.0 .OR. NO.GT.5) GOTO 1
0015      DO 8 IN=1,NO
0016      4 WRITE(5,5) IN, DOLST(IN)
0017      5 FORMAT(3X,'EB/NJ(''11,'[,F8.5,' dB]: :, $)
0018      6 READ(5,6,ERR=4) REPLY
0019      6 FORMAT(AB)
0020      IF(REPLY.EQ.BLANKS) THEN
DEBNOL(IN)=DOLST(IN)
0021      ELSE
0022      READ(REPLY,7,ERR=4) DEBNOL(IN)
0023      7 FORMAT(F8.5)
0024      END IF
0025      CONTINUE
0026      8 CONTINUE
0027      9 WRITE(5,10)
0028      10 FORMAT(1H HOW MANY GAMMA? [10] :, $)
0029      READ(5,11,ERR=9) NG
0030      11 FORMAT(12)
0031      11 IF(NG.EQ.0) NG=10
0032      11 IF(NG.LT.0 .OR. NG.GT.10) GOTO 9
0033      11 DO 15 IN=1,NG
0034      12 WRITE(5,13) IN, DGAM(IN)
0035      13 FORMAT(3X,'GAMMA(''12.'[,F5.3,': :, $)
0036      13 RLAD(5,14,ERR=12) GAMLST(IN)
0037      14 FORMAT(FB,6)
0038      14 IF(GAMLST(IN).EQ.0.D0) GAMLST(IN)=DGAM(IN)
0039      14 IF(GAMLST(IN).LE.0.D0 .OR. GAMLST(IN).GT.1.D0) GOTO 12
0040      15 CONTINUE
0041      16 WRITE(5,17)
0042      17 FORMAT(1H HOW MANY EB/NJ? [126] :, $)
0043      READ(5,18,ERR=16) NJ
0044      18 FORMAT(13)
0045      18 IF(NJ.EQ.0) NJ=126
```

```

PDP-11 FORTRAN-77 V4.0-1 09-50:17 17-Jul-86 Page 5
SELF12M2.FTN;13 /F77/TR:BLOCKS/MR

0001      SUBROUTINE PSUBE(IQ,PE)
          C COMPUTE ERROR PROBABILITY
          C
          C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          C PARAMETER (SLOTS=2400,DO)
          C PARAMETER (SLOTS1=399,DO)
          C DIMENSION WORK(20),STACK(20),HEAP(20)
          C EXTERNAL D616,F01A,F01B,F01C,F01D,F108,F10C
          C COMMON /ROSE/RHOM,RHOT,GAMMA,BIGK
          C COMPUTE ELEMENTAL EVENT PROBABILITIES
          C
          0=10
          P12=q*(Q-1.00)/SLOTPR
          P11=q*(SLOTS-Q)/SLOTPR
          P10=(SLOTS-Q)*SLOTPR
          PART=P10*P10*0.500*DEXP(-RHOM)*(1.00+RHOM/3.00)
          PB=PART
          CALL ADQUAD(0.00,1.00,PART,D616,F01A,1.0-9,WORK,STACK,HEAP,
                     20,KODE)
          IF(KODE.NE.0) STOP 'ADQUAD ERROR FIRST CALL'
          PART=2.00*P10*P11*PART
          PB=PB+PART
          CALL ADQUAD(0.00,1.00,PART,D616,F01B,1.0-9,WORK,STACK,HEAP,
                     20,KODE)
          IF(KODE.NE.0) STOP 'ADQUAD ERROR SECOND CALL'
          PART=P11*P11*PART
          PB=PB+PART
          CALL ADQUAD(0.00,1.00,PART,D616,F01A,1.0-9,WORK,STACK,HEAP,
                     20,KODE)
          IF(KODE.NE.0) STOP 'ADQUAD ERROR THIRD CALL'
          PART=2.00*P10*P11*PART
          PB=PB+PART
          RDTIFF=RHOM-RHOT
          RSUM=RHOM+RHOT
          PART=DEXP(-RHOT)*(RHOM+RDTIFF-RSUM)
          PAR=PAR+DEXP(-RHOM)*(RHOT+RDTIFF+RSUM)
          PART=PART/DX1(RDTIFF,3)
          PART=2.00*P10*P12*PART
          PB=PB+PART
          CALL ADQUAD(0.00,1.00,PART,D616,F01C,1.0-9,WORK,STACK,HEAP,
                     20,KODE)
          IF(KODE.NE.0) STOP 'ADQUAD ERROR FOURTH CALL'
          PART=2.00*P11*PART
          PB=PB+PART
          CALL ADQUAD(0.00,1.00,PART,D616,F01D,1.0-9,WORK,STACK,HEAP,
                     20,KODE)
          IF(KODE.NE.0) STOP 'ADQUAD ERROR FIFTH CALL'
          PART=2.00*P11*P12*PART
          PB=PB+PART

```

```

PDP-11 FORTRAN-77 V4.0-1 09:50:17 17-Jul-86 Page 6
SELF12M2.FTN;13 /F77/TR:BLOCKS/MR

0042      CALL ADQUAD(0.00,1.00,PART,D616,F108,1.0-9,WORK,STACK,HEAP,
                     20,KODE)
          IF(KODE.NE.0) STOP 'ADQUAD ERROR SIXTH CALL'
          PART=P11*P11*PART
          PB=PB+PART
          CALL ADQUAD(0.00,1.00,PART,D616,F10C,1.0-9,WORK,STACK,HEAP,
                     20,KODE)
          IF(KODE.NE.0) STOP 'ADQUAD ERROR SEVENTH CALL'
          PART=2.00*P11*P12*PART
          PB=PB+PART
          PART=0.500*DEXP(-RHOT)*(1.00+RHOT/3.00)
          PART=P12*P12*PART
          PB=PB+PART
          RETURN
          END

PDP-11 FORTRAN-77 V4.0-1 09:50:23 17-Jul-86 Page 7
SELF12M2.FTN;13 /F77/TR:BLOCKS/MR

0001      DOUBLE PRECISION FUNCTION P101(Z)
          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          COMMON /ROSE/RHOM,RHOT,GAMMA,BIGK
          D=1.00+(BIGK-1.00)*Z
          X=BIGK*RHOM*Z/0
          P101=BIGK*DEXP(X-RHOM)*(1.00+X)/(D*X)
          RETURN
          END

PDP-11 FORTRAN-77 V4.0-1 09:50:25 17-Jul-86 Page 8
SELF12M2.FTN;13 /F77/TR:BLOCKS/MR

0001      DOUBLE PRECISION FUNCTION P110(Z)
          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          COMMON /ROSE/RHOM,RHOT,GAMMA,BIGK
          D=BIGK-(BIGK-1.00)*Z
          X=RHOT*Z/0
          P110=BIGK*DEXP(X-RHOM)*(1.00+X)/(D*X)
          RETURN
          END

PDP-11 FORTRAN-77 V4.0-1 09:50:26 17-Jul-86 Page 9
SELF12M2.FTN;13 /F77/TR:BLOCKS/MR

0001      DOUBLE PRECISION FUNCTION F01AV()
          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          COMMON /ROSE/RHOM,RHOT,GAMMA,BIGK
          F01A=(1.00-V)*DEXP(-RHOM*V)*P101(V)
          RETURN
          END

```

```

PDP-11 FORTRAN-77 V4.0-1 09:50:27 17-Jul-86
SELF2M2.FTN;13 /F77/TR:BLOCKS/MR

0001    DOUBLE PRECISION FUNCTION F01B(V)
0002    IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0003    COMMON /ROSE/ RHON, RHOT, GAMMA, BIGK
0004    DMV=1.00-V
0005    D=V+BIGK*DMV
0006    F01B=P101(V)*BIGK*DEXP(-V*RHOM*D)+DMV/D
0007    RETURN
0008    END

```

Page 10

PDP-11 FORTRAN-77 V4.0-1 09:50:35 17-Jul-86
SELF2M2.FTN;13 /F77/TR:BLOCKS/MR

```

0001    DOUBLE PRECISION FUNCTION F10B(V)
0002    IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0003    COMMON /ROSE/ RHON, RHOT, GAMMA, BIGK
0004    DMV=1.00-V
0005    D=V+BIGK*DMV
0006    F01B=P101(V)*BIGK*DEXP(-V*RHOM*D)+DMV/D
0007    RETURN
0008    END

```

```

PDP-11 FORTRAN-77 V4.0-1 09:50:29 17-Jul-86
SELF2M2.FTN;13 /F77/TR:BLOCKS/MR

0001    DOUBLE PRECISION FUNCTION F10A(V)
0002    IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0003    COMMON /ROSE/ RHON, RHOT, GAMMA, BIGK
0004    F10A=(1.00-V)*DEXP(-RHOM*V)*P110(V)
0005    RETURN
0006    END

```

Page 11

PDP-11 FORTRAN-77 V4.0-1 09:50:37 17-Jul-86
SELF2M2.FTN;13 /F77/TR:BLOCKS/MR

```

0001    DOUBLE PRECISION FUNCTION F10C(V)
0002    IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0003    COMMON /ROSE/ RHON, RHOT, GAMMA, BIGK
0004    F10C=(1.00-V)*DEXP(-RHOT*V)*P110(V)
0005    RETURN
0006    END

```

Page 12

```

PDP-11 FORTRAN-77 V4.0-1 09:50:31 17-Jul-86
SELF2M2.FTN;13 /F77/TR:BLOCKS/MR

0001    DOUBLE PRECISION FUNCTION P110(V)
0002    IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0003    COMMON /ROSE/ RHON, RHOT, GAMMA, BIGK
0004    DMV=1.00-V
0005    D=BIGK*DMV+DMV
0006    P110=DMV*DEXP(-BIGK*RHOT*V/D)/D
0007    RETURN
0008    END

```

X-5

```

PDP-11 FORTRAN-77 V4.0-1 09:50:32 17-Jul-86
SELF2M2.FTN;13 /F77/TR:BLOCKS/MR

0001    DOUBLE PRECISION FUNCTION F01C(V)
0002    IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0003    COMMON /ROSE/ RHON, RHOT, GAMMA, BIGK
0004    F01C=P101(V)*P110(V)
0005    RETURN
0006    END

```

Page 13

PDP-11 FORTRAN-77 V4.0-1 09:50:34 17-Jul-86
SELF2M2.FTN;13 /F77/TR:BLOCKS/MR

```

0001    DOUBLE PRECISION FUNCTION F01D(V)
0002    IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0003    COMMON /ROSE/ RHON, RHOT, GAMMA, BIGK
0004    F01D=(1.00-V)*DEXP(-RHOT*V)*P101(V)
0005    RETURN
0006    END

```

Page 14

```

PDP-11 FORTRAN-77 V4.0-1 09:50:34 17-Jul-86
SELF2M2.FTN;13 /F77/TR:BLOCKS/MR

0001    DOUBLE PRECISION FUNCTION F01E(V)
0002    IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0003    COMMON /ROSE/ RHON, RHOT, GAMMA, BIGK
0004    F01E=(1.00-V)*DEXP(-RHOT*V)*P101(V)
0005    RETURN
0006    END

```

PDP-11 FORTRAN-77 V4.0-1 09:50:35 17-Jul-86
SELF2M2.FTN;13 /F77/TR:BLOCKS/MR

```

0001    DOUBLE PRECISION FUNCTION F10C(V)
0002    IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0003    COMMON /ROSE/ RHON, RHOT, GAMMA, BIGK
0004    DMV=1.00-V
0005    D=V+BIGK*DMV
0006    F10C=P101(V)*BIGK*DEXP(-V*RHOM*D)+DMV/D
0007    RETURN
0008    END

```

Page 15

PDP-11 FORTRAN-77 V4.0-1 09:50:37 17-Jul-86
SELF2M2.FTN;13 /F77/TR:BLOCKS/MR

```

0001    DOUBLE PRECISION FUNCTION F10D(V)
0002    IMPLICIT DOUBLE PRECISION(A-H,0-Z)
0003    COMMON /ROSE/ RHON, RHOT, GAMMA, BIGK
0004    DMV=1.00-V
0005    D=V+BIGK*DMV
0006    F10D=P101(V)*BIGK*DEXP(-V*RHOT*D)+DMV/D
0007    RETURN
0008    END

```

Page 16

```

SUBROUTINE ADQUAD(XL,XU,Y,QR,F,TOL,WORK,STACK,HEAP,N,KODE)
C
C   ADAPTIVE QUADRATURE ALGORITHM
C   XL - LOWER LIMIT OF INTEGRAL (IN)
C   XU - UPPER LIMIT OF INTEGRAL (IN)
C   Y - VALUE OF INTEGRAL (OUT)
C   QR - NAME OF A QUADRATURE RULE SUBROUTINE (IN)
C
C   WITH CALLING SEQUENCE
C     CALL QR(XL,XU,F,Y)
C
C   F - NAME OF FUNCTION TO BE INTEGRATED (IN)
C   TOL - ERROR TOLERANCE FOR FINAL ANSWER (IN)
C   WORK - WORK ARRAY OF SIZE N (IN)
C   STACK - SECOND WORK ARRAY OF SIZE N, MUST NOT BE
C   HEAP - THIRD WORK ARRAY, SIZE N, DISTINCT FROM WORK AND STACK
C   SAME ARRAY AS WORK (IN)
C   N - SIZE OF WORK AND STACK; MAX. NO. OF BISECTIONS (IN)
C   KODE - ERROR INDICATOR (OUT)
C     0... NO ERROR
C     1... WORK ARRAYS TOO SMALL
C     2... EPS DIVIDED TO ZERO, EITHER ASKING FOR TOO
C          TIGHT A TOLERANCE OR ROUND-OFF PREVENTS
C          ATTAINING REQUIRED ACCURACY
C
C   C. R. H. FRENCH, 14 AUGUST 1984
C
C-6 0002
C-6 0003
C-6 0004
C-6 0005
C-6 0006
C-6 0007
C-6 0008
C-6 0009
C-6 0010
C-6 0011
C-6 0012
C-6 0013
C-6 0014
C-6 0015
C-6 0016
C-6 0017
C-6 0018
C
C   IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C   EXTERNAL F
C   DIMENSION WORK(N),STACK(N),HEAP(N)
C   KODE=0
C   Y=0.0G
C   WORK(1)=XU
C   CALL QR(XL,XU,F,T)
C   HEAP(1)=T
C   A=XL
C   NPTS=1
C   EPS=TOL
C   STACK(1)=EPS
C   B=WORK(NPTS)
C   XH=(A+B)*0.5D0
C   CALL QR(A,XH,F,P1)
C   CALL QR(XH,B,F,P2)
C   IF(DABS(T-P1-P2).LE.EPS) GOTO 20
C
C   SPLIT IT
C   NPTS=NPTS+1
C   IF(NPTS.GT.N) THEN
C     KODE=1
C   RETURN
C
C-6 0019
C-6 0020
C-6 0021
C-6 0022
C-6 0023
C-6 0024
C-6 0025
C-6 0026
C-6 0027

```

PDP-11 FORTRAN-77 V4.0-1 09:50:41 17-Jul-86
SELF2M2.FTN;13 /F77/TR:BLOCKS/MR

Page 19

```
0001      SUBROUTINE DG16(A,B,F,ANSWER)
C   16-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL
C   REF.: ABRAMOWITZ & STEGUN, FN. 25.4.30 AND TABLE 25.4
C   R. H. FRENCH, 28 FEBRUARY 1986
C   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0002      DIMENSION X(8),W(8)
C   WEIGHTS AND ABSCISSAS FOR 16-POINT GAUSSIAN QUADRATURE
C
0003      DATA X/ 0.095012509837637440186500,
C           0.28160355077925691323000,
C           0.45801577765722738634200,
C           0.6178762440264374844700,
C           0.755404408355000303309500,
C           0.865631202388783174388000,
C           0.94457502307323257607800,
C           0.98940093499164993259600 /
0004      DATA W/ 0.18945061045506849625500,
C           0.18260341504492358886700,
C           0.16915651939500253818900,
C           0.14959598881657673208100,
C           0.1246289712553387205200,
C           0.09515851168249278481000,
C           0.06226352393864789286300,
C           0.02715245941175409485200 /
ANSWER=0.00
BMA02=(B-A)/2.00
BPA02=(B+A)/2.00
DO 10 I=1,8
C=X(I)*BMA02
Y1=BPA02+C
Y2=BPA02-C
ANSWER=ANSWER+W(I)*(F(Y1)+F(Y2))
CONTINUE
10 ANSWER=ANSWER*BMA02
RETURN
END
```


J. S. LEE ASSOCIATES, INC.

APPENDIX L COMPUTER PROGRAM FOR SELF-NORMALIZING RECEIVER WITH M=2, L=3

The following pages contain the source code listing for the FORTRAN-77 program used for numerical computations of the performance of the self-normalizing receiver when $M=2$ and $L=3$. The program searches numerically for the worst-case jamming fraction.



```

0001      PROGRAM SELF23
          C
          C SELF NORMALIZING RECEIVER, N=2, L=3
          C
          C ANALYSIS: L.E. MILLER
          C PROGRAM: R.H. FRENCH
          C
          C METHOD: SELF-COMVOLUTION OF THE MARGINAL DENSITY
          C
          C IMPLICIT DOUBLE PRECISION (A-H,0-7)
          C PARAMETER(M=2, L=3, NSLOTS=2400, SLOTS=2400.00)
          C LOGICAL TEST
          C
          0002      CHARACTER*13 FRAME
          0003      REAL DEBNJ(126), PELOG(126), QOPT(126)
          0004      PARAMETER(SLOTPR=5757600.00)
          0005      LOGICAL TEST
          0006      CHARACTER*13 FRAME
          0007      REAL DEBNJ(126), PELOG(126), QOPT(126)
          0008      VIRTUAL PEOLD(2400)
          0009      COMMON /PIES/ P10, P11, P12
          0010      COMMON /ROSE/ RHOM, RHOT, GAMMA, BIGK, BIGK1
          0011      COMMON /RONCOM/ A0N, A1N, A2N, A3N, A0T, A1T, A2T, A3T,
          0012      COMMON /PARMS/ NO, NJ, DEBNL(5),
          0013      CALL ERSET(20,.TRUE.,.FALSE.,.FALSE.,.15)
          0014      CALL GET
          0015      DC 900 10-1, NO
          0016      EBNO=10.00**((DEBNL(10)/10.00)
          0017      RHON=EBNO/L
          0018      A0N=1.0D0*RHOM*RHOM*RHON/6.00
          0019      A1N=RHOM*RHON/12.00-1.00
          0020      A2N=-RHOM*(1.00+RHOM/2.00)
          0021      A3N=-RHOM*RHON/6.00
          0022      100UT=DEBNL(10)
          C
          C PROGRESS FILE
          C
          0023      READ(4) EBNOIN, DBJOIN, D1IN
          0024      1 FORMAT('S23',12.2,'GOPT.DAT')
          0025      WRITE(15,2) FNAME
          0026      2 FORMAT(12X,FILE-'A13')
          0027      OPEN(UNIT=4,FILE=FNAME,STATUS='OLD',ERR=810)
          0028      FORM='UNFORMATTED',ACCESS='SEQUENTIAL'
          C
          C HAVE A PROGRESS FILE, READ IT
          C
          0029      READ(4,END=802) DEBNJ(JJ). PELOG(JJ), QOPT(JJ)
          0030      JJ=0
          0031      JJ=JJ+
          0032      READ(4,END=802) DEBNJ(JJ). PELOG(JJ), QOPT(JJ)
          0033      GOTO 801
          0034      CLOSE(UNIT=4)
          0035      GOTO 802

```

```

PDP-11 FORTRAN-77 V4.0-1 10:01:40 17-Jul-86 Page 3 PDP-11 FORTRAN-77 V4.0-1 10:01:40 17-Jul-86 Page 4
SELF3M2B.FTN;16 /F77/MR SELF3M2B.FTN;16 /F77/MR

0079 C=RHOT+RHOT*(RHOM-RHOT)+2.D0*RHOM*RHOT
0080 C1=RHOM*(RHOM-RHOT)*(RHOT+RHOM+RHOT-RHOM)
0081 CALL PSUBTPE,PELOG,Q
0082 P3=P2
0083 IF(P3.GT.P2 .AND. 10.LT.NSLOTS) THEN
0084 C KEEP ON GOING, WE ARE NOT PAST THE MAXIMUM
0085 P1=P2
0086 P2=P3
0087 IQ=MIMO(10*IDQ,NSLOTS)
0088 Q=DMIN1(Q+DQ,NSLOTS)
0089 GOTO 709
0090 PMAX=DMAX1(P1,P2,P3)
0091 EPS=0.001D0*PMAX
0092 TEST=(DABS(P1-P2).LE.EPS .AND. DABS(P1-P3).LE.EPS .AND.
0093 C IF(TEST .OR. IDQ.EQ.1
0094 C .OR. ((.NOT.TEST) .AND. 1Q.EQ.NSLOTS)) THEN
0095 C WE ARE DONE WHEN ALL 3 ARE CLOSE TOGETHER OR WHEN DQ=1
0096 C OR WHEN WE REACHED FULL-BAND JAMMING AND P(E) IS STILL
0097 C INCREASING
0098 C PREVENT ROUND-OFF FROM MAKING QOPT VS. EB/NJ NON-MONOTONIC
0099 IF((QOPT(IJ).LT.QOPT(IJ-1)) QOPT(IJ)=QOPT(IJ-1)
0100 END IF
0101 ELSE
0102 C THE OPTIMUM IS FULL-BAND JAMMING
0103 QOPT(IJ)=NSLOTS
0104 END IF
0105 C GOTO 665
0106 ELSE
0107 C NOT LOCATED SUFFICIENTLY ACCURATELY, CUT DQ AND TRY AGAIN
0108 Q=Q-DQ
0109 DQ=IDQ-1DQ
0110 IDQ=IDQ/2
0111 P1=0.D0
0112 P2=0.D0
0113 P3=0.D0
0114 GOTO 709
0115 END IF
0116 END IF
0117 665 PELOG(IJ)=NLOG10(POPT)
0118 OPEN(UNIT=4,FILE=FNAME,STATUS='OLD',ACCESS='APPEND',
0119      FORM='UNFORMATTED')
0120 CLOSE(UNIT=4)
0121 CONTINUE
0122 OPEN(UNIT=4,FILE=FNAME,STATUS='NEW',ACCESS='SEQUENTIAL',
0123      FORM='UNFORMATTED')
0124 WRITE(4,M,1,DEBNL(10),MSLOTS,MJ,DEBNJ,PELOG,OPT)
0125 WRITE(6,710) M,L,DEBNL(10),MSLOTS,
0126      (DEBNJ(IJ),10.D0+PELOG(IJ),1J=1,MJ)
0127 5X,'EB/NO = ',FB,5,'dB',5X,'MSLOTS = ',I4/
0128      ,EB/NJ (dB),8X,P(E),8X,OPT/<ND>(4X,OPT4.1,8X,1PD10.3,
0129      3X,OPT5.0))
0130 CONTINUE
0131 800 CONTINUE
0132 900 STOP 'PLEASE PURGE S23*.DAT'
0133 END

```

PDP-11 FORTRAN-77 V4.0-1 10:01:57 17-Jul-86

SELF1.F90,FTN;16 /F77/MR

Page 5

PDP-11 FORTRAN-77 V4.0-1 /F77/MR 10:02:02 17-Jul-86

SELF1.F90,FTN;16 /F77/MR

Page 6

```
0001      SUBROUTINE GET
          C INTERACTIVE RUN PARAMETER INPUTS
          C
          C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          C DIMENSION DOLST(5)
          C CHARACTER*8 REPLY, BLANKS
          C COMMON /PARMS/ NO, NJ, DEBNDL(5),
          C                  DEBNR(126), DJ, DSJ0
          C DATA DOLST/ 13.3524700, 12.313300, 10.9444300, 14.8925300,
          C             16.02713500/
          C DATA BLANKS/./
          C
          C WRITE(5,2)
          C FORMAT(' HOW MANY EB/NJ? [1] ',\$)
          C READ(5,3,ERR=1) NO
          C 3 FORMAT([1])
          C IF(NO.EQ.0) NO=1
          C IF((NO.LT.0 .OR. NO.GT.5) GOTO 1
          C DO & IN=1,NO
          C   WRITE(5,5) IN, DOLST(IN)
          C   FORMAT(3X,'EB/ND( ','IN,)' [,F6.5,' dB]: ',\$)
          C   READ(5,6,ERR=4) REPLY
          C   FORMAT(AB8)
          C   IF(REPLY.EQ.BLANKS) THEN
          C     DEBNDL(IN)=DOLST(IN)
          C   ELSE
          C     READ(REPLY,7,ERR=4) DEBNDL(IN)
          C   FORMAT(F8.5)
          C   END IF
          C 4 CONTINUE
          C 5 WRITE(5,17)
          C 6 FORMAT(' HOW MANY EB/NJ? [5] ',\$)
          C 7 READ(5,18,ERR=16) NJ
          C 8 FORMAT(13)
          C 9 IF(NJ.EQ.0) NJ=5
          C 10 IF((NJ.LT.0 .OR. NJ.GT.126) GOTO 16
          C 11 WRITE(5,20)
          C 12 FORMAT(13)
          C 13 READ(5,21,ERR=19) DSJ0
          C 14 FORMAT(F5.2)
          C 15 IF(DSJ0.EQ.0.00) DSJ0=50.00
          C 16 DJ=0.00
          C 17 IF(NJ.GT.1) THEN
          C 18 WRITE(5,23)
          C 19 FORMAT(' INCREMENT FOR EB/NJ [-1 dB]: ',\$)
          C 20 READ(5,24,ERR=22) DJ
          C 21 FORMAT(F5.2)
          C 22 IF(DJ.EQ.0.00) DJ=-1.00
          C 23 END IF
          C 24 RETURN
          C 25 END
```

PDP-11 FORTRAN-77 V4.0-1 /F77/MR
SELF32B.FTN16

Page 7

PDP-11 FORTRAN-77 V4.0-1 /F77/MR
SELF32B.FTN16

Page 8

0001 5 SUBROUTINE ADQUA(XL,XU,Y,OR,F,TOL,ABSTOL,
 WORK,STACK,HEAP,N,KODE)
C
C ADAPTIVE QUADRATURE ALGORITHM
C XL - LOWER LIMIT OF INTEGRAL (IN)
C XU - UPPER LIMIT OF INTEGRAL (IN)
C Y - VALUE OF INTEGRAL (OUT)
C OR - NAME OF A QUADRATURE RULE SUBROUTINE (IN)
C WITH CALLING SEQUENCE
C CALL OR(XL,XU,F,Y)
C F - NAME OF FUNCTION TO BE INTEGRATED (IN)
C TOL - ERROR TOLERANCE FOR FINAL ANSWER (IN)
C ABSTOL - ABSOLUTE ERROR TOLERANCE (IN)
C WORK - WORK ARRAY OF SIZE N (IN)
C STACK - SECOND WORK ARRAY OF SIZE N, MUST NOT BE
 HEAP- THIRD WORK ARRAY, SIZE N, DISTINCT FROM WORK AND STACK
C SAME ARRAY AS WORK (IN)
C N - SIZE OF WORK AND STACK; MAX. NO. OF BISECTIONS (IN)
C KODE - ERROR INDICATOR (OUT)
C 0 -- NO ERROR
C 1 -- WORK ARRAYS TOO SMALL
C R. H. FRENCH, 14 AUGUST 1984

 5.0002 IMPLICIT DOUBLE PRECISION(A-H,O-Z)
 5.0003 EXTERNAL F
 5.0004 DIMENSION WORK(N),STACK(N),HEAP(N)
 5.0005 KODE=0
 5.0006 Y=0.00
 5.0007 WORK(1)=XU
 5.0008 CALL OR(XL,XU,F,1)
 5.0009 HEAP(1)=T
 5.0010 A=XL
 5.0011 NPTS=1
 5.0012 EPS=TOL
 5.0013 STACK(1)=EPS
 5.0014 B=HORC(NPTS)
 5.0015 XH=(A+B)*0.500
 5.0016 CALL OR(A,XH,F,P1)
 5.0017 CALL OR(XH,B,F,P2)
 5.0018 TEST=MAX1(EPS,DABS(T),ABSTOL)
 5.0019 IF(DABS(T-P1-P2).LE.TEST .OR. DABS(T).LE.ABSTOL) GOTO 20
 C SPLIT IT
 NPTS=NPTS+1
 IF(NPTS.GT.N) THEN
 KODE=1
 RETURN
 END IF
 WORK(NPTS)=XM
 HEAP(NPTS)=P2
 T=P1
 EPS=MAX1(EPS/2.00,5.D-16)

PDP-11 FORTRAN-77 V4.0-1 10:02:06 17-Jul-86
SELFL342B.FTM;16 /F77/MR

Page 9

PDP-11 FORTRAN-77 V4.0-1 10:02:06 17-Jul-86
SELFL342B.FTM;16 /F77/MR

0001 S SUBROUTINE ADQUA2(XL,XU,Y,QR,F,TOL,ABSTOL,
 WORK,STACK,HEAP,N,KODE)

C ADAPTIVE QUADRATURE ALGORITHM

C XL - LOWER LIMIT OF INTEGRAL (IN)

C XU - UPPER LIMIT OF INTEGRAL (IN)

C Y - VALUE OF INTEGRAL (OUT)

C QR - NAME OF A QUADRATURE RULE SUBROUTINE (IN)

C WITH CALLING SEQUENCE
CALL QR(XL,XU,F,Y)

C F - NAME OF FUNCTION TO BE INTEGRATED (IN)

C TOL - ERROR TOLERANCE FOR FINAL ANSWER (IN)

C ABSTOL - ABSOLUTE ERROR TOLERANCE (IN)

C WORK - WORK ARRAY OF SIZE N (IN)

C STACK - SECOND WORK ARRAY OF SIZE N. MUST NOT BE

C HEAP - THIRD WORK ARRAY. SIZE N, DISTINCT FROM WORK AND STACK

C SAME ARRAY AS WORK (IN)

C N - SIZE OF WORK AND STACK; MAX. NO. OF BISECTIONS (IN)

C KODE - ERROR INDICATOR (OUT)

C 0 -- NO ERROR

C 1 -- WORK ARRAYS TOO SMALL

C R. H. FRENCH, 14 AUGUST 1984

C

 IMPLICIT DOUBLE PRECISION(A-H,O-Z)

EXTERNAL F

DIMENSION WORK(N),STACK(N),HEAP(N)

KODE=0

Y=0.00

WORK(1)=XU

CALL QR(XL,XU,F,T)

HEAP(1)=T

A=XL

NPTS=1

EPS=TOL

STACK(1)=EPS

B=WORK(NPTS)

XH=(A+B)*0.500

CALL QR(A,XH,F,P1)

CALL QR(XH,B,F,P2)

TEST=DMAX1(EPS,DABS(T),ABSTOL)

IF(DABS(T-P1-P2).LE.TEST .OR. DABS(T).LE.ABSTOL) GOTO 20

C SPLIT IT

NPTS=NPTS+1

IF(NPTS.GT.N) THEN

KODE=1

RETURN

END IF

WORK(NPTS)=XH

HEAP(NPTS)=P2

T=P1

EPS=DMAX1(EPS/2.00,5.0-16)

PDP-11 FORTRAN-77 V4.0-1 10:02:06 17-Jul-86
SELFL342B.FTM;16 /F77/MR

PDP-11 FORTRAN-77 V4.0-1 10:02:06 17-Jul-86
SELFL342B.FTM;16 /F77/MR

0029 STACK(NPTS)=EPS

0030 GOTO 10

 C FINISHED A PIECE

0031 20 Y=Y+P1+P2

0032 EPS=STACK(NPTS)

0033 T=HEAP(NPTS)

0034 NPTS=NPTS-1

0035 A=8

0036 IF(NPTS.EQ.0) RETURN

0037 GOTO 10

0038 END

PDP-11 FORTRAN-77 V4.0-1 10:02:10 17-Jul-86 Page 11

SELF13M28.FTN;16 /F77/MR

0001 5

 SUBROUTINE ADQUA3(XL,XU,Y,QR,F,TOL,ABSTOL,
 WORK,STACK,HEAP,N,KODE)

 C ADAPTIVE QUADRATURE ALGORITHM
 C XL - LOWER LIMIT OF INTEGRAL (IN)
 C XU - UPPER LIMIT OF INTEGRAL (IN)
 C Y - VALUE OF INTEGRAL (OUT)
 C QR - NAME OF A QUADRATURE RULE SUBROUTINE (IN)
 C WITH CALLING SEQUENCE
 C CALL QR(XL,XU,F,Y)
 C F - NAME OF FUNCTION TO BE INTEGRATED (IN)
 C TOL - ERROR TOLERANCE FOR FINAL ANSWER (IN)
 C ABSTOL - ABSOLUTE ERROR TOLERANCE (IN)
 C WORK - WORK ARRAY OF SIZE N (IN)
 C STACK - SECOND WORK ARRAY OF SIZE N, MUST NOT BE
 C HEAP - THIRD WORK ARRAY, SIZE N, DISTINCT FROM WORK AND STACK
 C SAME ARRAY AS WORK (IN)
 C N - SIZE OF WORK AND STACK; MAX. NO. OF BISECTIONS (IN)
 C KODE - ERROR INDICATOR (OUT)
 C 0 -- NO ERROR
 C 1 -- WORK ARRAYS TOO SMALL
 C R. H. FRENCH, 14 AUGUST 1984

 C IMPLICIT DOUBLE PRECISION(A-H,O-Z)

 C EXTERNAL F
 C DIMENSION WORK(N),STACK(N),HEAP(N)
 C KODE=0
 C Y=0.00
 C WORK(1)=XU
 C CALL QR(XL,XU,F,T)
 C HEAP(1)=T
 C A=XL
 C NPTS=1
 C EPS=TQ
 C STACK(1)=EPS
 C B=MWORK(NPTS)
 C XM=(A+B)*0.500
 C CALL QR(XM,F,P1)
 C CALL QR(XM,B,F,P2)
 C TEST=MAX(EPS,DABS(T).ABSTOL)
 C IF(DABS(T-P1-P2).LE.TEST .OR. DABS(T).LE.ABSTOL) GOTO 20
 C SPLIT IT
 C NPTS=NPTS+1
 C IF(NPTS.GT.N) THEN
 C KODE=1
 C RETURN
 C END IF
 C WORK(NPTS)=XM
 C HEAP(NPTS)=P2
 C T=P1
 C EPS=DMAX1(EPS/2.00,5.D-16)

PDP-11 FORTRAN-77 V4.0-1 10:02:10 17-Jul-86 Page 12

SELF13M28.FTN;16 /F77/MR

0029 STACK(NPTS)=EPS
 GOTO 10
 C FINISHED A PIECE
 20 Y=Y+P1+P2
 EPS=STACK(NPTS)
 T=HEAP(NPTS)
 NPTS=NPTS-1
 A=B
 IF(NPTS.EQ.0) RETURN
 GOTO 10
 END

PDP-11 FORTRAN-77 V4.0-1 10:02:13 17-Jul-86 Page 13

SELF13M28.FTN;16 /F77/MR

0030 BLOCK DATA
 IMPLICIT DOUBLE PRECISION(A-H,O-Z)
 COMMON /GQNTS/ X(5),W(5)
 C WEIGHTS AND ABSISSAS FOR 10-POINT GAUSSIAN QUADRATURE
 C

0031 DATA X / 0.14887433890163100,
 / 0.43339539412924700,
 / 0.579404956829902400,
 / 0.86506336668898500,
 / 0.97390652851717200 /
 / DATA W / 0.295552422471475300,
 / 0.26926671939999500,
 / 0.21908636251598200,
 / 0.14945134915058100,
 / 0.086667134430868800 /
 / END

0032 10

0033 11

0034 12

0035 13

0036 14

0037 15

0038 16

PDP-11 FORTRAN-77 V4.0-1 /F77/MR
SELF13M2B.FTN;16

Page 14 PDP-11 FORTRAN-77 V4.0-1 10:02:14 17-Jul-86 Page 16

0001 SUBROUTINE DS10A(A,B,F,ANSWER)
C 10-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL
C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4
C R. H. FRENCH, 30 MAY 1986

0002 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON /GQNTS/ X(5),W(5)
ANSWER=0.0D
0003 BMA02=(B-A)/2.0D
0004 BPA02=(B+A)/2.0D
0005 DO 10 I=1,5
0006 C=X(I)*BMA02
0007 Y1=BPA02+C
0008 Y2=BPA02-C
0009 ANSWER=ANSWER+W(I)*(F(Y1)+F(Y2))
0010 CONTINUE
0011 10 ANSWER=ANSWER*BMA02
0012 RETURN
0013 END

0001 SUBROUTINE DS10B(A,B,F,ANSWER)
C 10-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL
C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4
C R. H. FRENCH, 30 MAY 1986

0002 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON /GQNTS/ X(5),W(5)
ANSWER=0.0D
0003 BMA02=(B-A)/2.0D
0004 BPA02=(B+A)/2.0D
0005 DO 10 I=1,5
0006 C=X(I)*BMA02
0007 Y1=BPA02+C
0008 Y2=BPA02-C
0009 ANSWER=ANSWER+W(I)*(F(Y1)+F(Y2))
0010 CONTINUE
0011 10 ANSWER=ANSWER*BMA02
0012 RETURN
0013 END

PDP-11 FORTRAN-77 V4.0-1 /F77/MR
SELF13M2B.FTN;16

Page 15 PDP-11 FORTRAN-77 V4.0-1 10:02:16 17-Jul-86 Page 17

0001 SUBROUTINE DS10C(A,B,F,ANSWER)
C 10-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL
C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4
C R. H. FRENCH, 30 MAY 1986

0002 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON /GQNTS/ X(5),W(5)
ANSWER=0.0D
0003 BMA02=(B-A)/2.0D
0004 BPA02=(B+A)/2.0D
0005 DO 10 I=1,5
0006 C=X(I)*BMA02
0007 Y1=BPA02+C
0008 Y2=BPA02-C
0009 ANSWER=ANSWER+W(I)*(F(Y1)+F(Y2))
0010 CONTINUE
0011 10 ANSWER=ANSWER*BMA02
0012 RETURN
0013 END

0001 SUBROUTINE DS10B(F77/MR
SELF13M2B.FTN;16

Page 16 PDP-11 FORTRAN-77 V4.0-1 10:02:20 17-Jul-86 Page 17

0002 DOUBLE PRECISION FUNCTION PDF(Z)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
PARAMETER (CUSP=1.00)
EXTERNAL DS10B, CONV02
DIMENSION WORK(20), STACK(20), HEAP(20)
COMMON /PASSZ/ ZEE
ZEE=2
0003 XL=DMAX1(0,DO_Z-1.00)
XU=DMIN1(2,DO_Z)
0004 IF(XL.GE.CUSP.OR.XU.LE.CUSP) THEN
0005 CALL ADQUA2(XL,XU,PDF,DS10B,CONV02,1,D-4,1,L-10,
0006 WORK,STACK,HEAP,KODE)
0007 ELSE
0008 CALL ADQUA2(XL,CUSP,PX,DS10B,CONV02,1,D-4,1,D-10,
0009 WORK,STACK,HEAP,20,KODE)
0010 IF(KODE.NE.0) STOP 'ADQUA2-1 ERROR'
0011 CALL ADQUA2(CUSP,XU,PY,DS10B,CONV02,1,D-4,1,D-10,
0012 WORK,STACK,HEAP,20,KODE)
0013 IF(KODE.NE.0) STOP 'ADQUA2-2 ERROR'
0014 PDF=PX+PY
0015 END IF
0016 RETURN
0017 END
0018 0019
0020 0021

```

PDP-11 FORTRAN-77 V4.0-1 10:02:22 17-Jul-86
SELFLOAT.FTN;16 /F77/MR Page 18

0001      DOUBLE PRECISION FUNCTION CONV02(W)
0002      IMPLIT DOUBLE PRECISION(A-H,O-Z)
0003      EXTERNAL DG10C, F0001, F0010, F0110, F0111, F1011
      S
0004      DIMENSION WORK(20), STACK(20), HEAP(20)
      S
0005      COMMON /PASSZ/ ZEE
0006      COMMON /ROSE/ RHON, RHOH, GAMMA, BIGK, BIGK1
0007      COMMON /ROMCON/ A0M, A1N, A2N, A3N, A0T, A1T, A2T, A3T,
      S
0008      COMMON /ROMCON/ B0, B1, C0, C1
      S
0009      COMMON /PIES/ P10, P11, P12
      S
0010      DUEBEYU
      S
0011      RZN=RHON*W
      S
0012      RZT=RHOT*W
      S
0013      R12=RHON-RHOT
      S
0014      RSUM=RHOH+RHON
      S
0015      TRNT=2.0D0+RHON+RHOT
      S
0016      IF(W.LT.1.0D0) THEN
      S
0017          P2=P10*P10*(DEP(RZN-2.0D0+RHON))*W*(1.0D0+RZN+RHON*RZN/6.
      S
              +P12*P12*(DEP(RZT-2.0D0+RHOT))*W*(1.0D0+RZT+RZT+RZT/6.
      S
              +2.0D0*P10*P12*(DEP(RZN-RSUM)*(RHON*RHON*R12*W-TRNT
      S
              +DEP(RZT-RSUM)*(RHOT*R12*W+TRNT
      S
              DX1(R12,3))
      S
      ELSE
      S
0018          ZM1=W-1.0D0
      S
0019          ZM2=W-2.0D0
      S
0020          P2=P10*P10*(DEP(RZN-2.0D0+RHON))
      S
              *(A0M+(A1N+(A2N+A3N)*ZM1)*ZM1)*ZM1)
      S
              +P12*P12*(DEP(RZT-2.0D0+RHOT)
      S
              *(A0T+(A1T+(A2T+A3T)*ZM1)*ZM1)*ZM1);
      S
              +2.0D0*P10*P12*(DEP(RHOT*ZM2)*(B0+B1*ZM1)
      S
              +DEP(RHON*ZM2)*(C0+C1*ZM1))/DX1(R12,
      S
      END IF
      S
0022          YL=DMAX1(0.0D0,W-1.0D0)
      S
0023          YU=DMIN1(1.0D0,W)
      S
0024          IF(P10.NE.0.0D0 .AND. P11.NE.0.0D0) THEN
      S
0025              CALL ADQUA3(YL,YU,PART,DG10C,F0001,1,D-6,1,0-11,
      S
                  WORK,STACK,HEAP,20,KODE)
      S
0026          IF(KODE.NE.0) STOP 'ADQUA3-1'.
      S
0027          P2=P2+2.0D0*P10*P11*PART
      S
0028          CALL ADQUA3(YL,YU,PART,DG10C,F0010,1,D-6,1,0-11,
      S
0029          WORK,STACK,HEAP,20,KODE)
      S
0030          IF(KODE.NE.0) STOP 'ADQUA3-2'.
      S
0031          P2=P2+2.0D0*P10*P11*PART
      S
0032          END IF
      S
0033          IF(P11.NE.0.0D0) THEN
      S
0034              CALL ADQUA3(YL,YU,PART,DG10C,F0101,1,0-6,1,0-11,
      S
                  WORK,STACK,HEAP,20,KODE)
      S
0035          IF(KODE.NE.0) STOP 'ADQUA3-3'.
      S
0036          P2=P2+2.0D0*P11*PART
      S
0037          CALL ADQUA3(YL,YU,PART,DG10C,F0110,1,D-6,1,0-11,
      S
                  WORK,STACK,HEAP,20,KODE)

```

```

PPR-11 FORTRAN-77 V4.0-1 10:02:22 17-Jul-96
SELF13M2B.FTN;16 /F77/WR Page 19

0038 IF(KODE,NE.0) STOP 'ADQUA3-4'
0039 P2=P2+P11*P11*PART
0040 CALL ADQUA3(YU,YU,PART,0610C,F01101.1.D-6,1.D-11,
0041 WORK,STACK,HEAP,20,KODE)
0042 IF(KODE,NE.0) STOP 'ADQUA3-6'
0043 P2=P2+P11*P11*PART
0044 END IF
0045 IF(P11,NE.0.DD,AND.,PJ12,NE.0.DD) THEN
0046 CALL ADQUA3(YU,YU,PART,0610C,F0111.1.D-6,1.D-11,
0047 WORK,STACK,HEAP,20,KODE)
0048 IF(KODE,NE.0) STOP 'ADQUA3-5'
0049 P2=P2+2.0D+P11*P12*PART
0050 CALL ADQUA3(YU,YU,PART,0610C,F0111.1.D-6,1.D-11,
0051 WORK,STACK,HEAP,20,KODE)
0052 IF(KODE,NE.0) STOP 'ADQUA3-7'
0053 P2=P2+2.0D+P11*P12*PART
0054 END IF
0055 CONVO2=P2*P1(ZER-W)
0056 RETURN
0057 END

PPR-11 FORTRAN-77 V4.0-1 10:02:30 17-Jul-96
SELF13M2B.FTN;16 /F77/WR Page 20

0001 DOUBLE PRECISION FUNCTION P1(Z)
0002 IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003 COMMON /ROSE/RHOM,RHOT,GAMMA,BIGK,BIGK1
0004 COMMON /PIES/P10,P11,P12
0005 RH2=RHOM**2
0006 RT2=RHOT**2
0007 P1=P10*DEXP(RN2-RHOM)*(1.0D+RN2)
0008 BK12=BIGK1*D
0009 D=1.0D+BK12
0010 A=BIGK*RN2/D
0011 P01=BIGK*DEXP(A-RHOM)*((1.0D+A)/D)/D
0012 D=BIGK-BK12
0013 A=RTZ/D
0014 P1=P1+P11*((P01+BIGK*DEXP(A-RHOT))*(1.0D+A)/D)/D
0015 P1=P1+P12*DEXP(RTZ-RHOT)*(1.0D-RTZ)
0016 RETURN
0017 END

PPR-11 FORTRAN-77 V4.0-1 10:02:32 17-Jul-96
SELF13M2B.FTN;16 /F77/WR Page 21

0001 DOUBLE PRECISION FUNCTION F0001(Z)
0002 IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003 COMMON /PASSW/DUBEWU
0004 F0001=P01(Z)*P01(DUBEWU-Z)
0005 RETURN

```

```

PDP-11 FORTRAN-77 V4.0-1 10:02:34 17-Jul-86 Page 22
SELF3M2B.FTN;16 /F77/MR

0001 DOUBLE PRECISION FUNCTION F010(Z)
0002 IMPLICIT DOUBLE PRECISION (A-H,0-Z)
0003 COMMON /PASSW/ DUBEYU
0004 F010=P00(Z)*P10(DUBEYU-Z)
0005 RETURN
0006 END

```

```

PDP-11 FORTRAN-77 V4.0-1 10:02:35 17-Jul-86 Page 23
SELF3M2B.FTN;16 /F77/MR

0001 DOUBLE PRECISION FUNCTION F010(Z)
0002 IMPLICIT DOUBLE PRECISION (A-H,0-Z)
0003 COMMON /PASSW/ DUBEYU
0004 F010=P01(Z)*P10(DUBEYU-Z)
0005 RETURN
0006 END

```

```

PDP-11 FORTRAN-77 V4.0-1 10:02:37 17-Jul-86 Page 24
SELF3M2B.FTN;16 /F77/MR

0001 DOUBLE PRECISION FUNCTION F010(Z)
0002 IMPLICIT DOUBLE PRECISION (A-H,0-Z)
0003 COMMON /PASSW/ DUBEYU
0004 F010=P01(Z)*P10(DUBEYU-Z)
0005 RETURN
0006 END

```

```

PDP-11 FORTRAN-77 V4.0-1 10:02:36 17-Jul-86 Page 25
SELF3M2B.FTN;16 /F77/MR

0001 DOUBLE PRECISION FUNCTION F0111(Z)
0002 IMPLICIT DOUBLE PRECISION (A-H,0-Z)
0003 COMMON /PASSW/ DUBEYU
0004 F0111=P01(Z)*P11(DUBEYU-Z)
0005 RETURN
0006 END

```

```

PDP-11 FORTRAN-77 V4.0-1 10:02:40 17-Jul-86 Page 26
SELF3M2B.FTN;16 /F77/MR

0001 DOUBLE PRECISION FUNCTION F1010(Z)
0002 IMPLICIT DOUBLE PRECISION (A-H,0-Z)
0003 COMMON /PASSW/ DUBEYU
0004 F1010=P00(Z)*P10(DUBEYU-Z)
0005 RETURN
0006 END

```

```

PDP-11 FORTRAN-77 V4.0-1 10:02:41 17-Jul-86 Page 27
SELF3M2B.FTN;16 /F77/MR

0001 DOUBLE PRECISION FUNCTION F1011(Z)
0002 IMPLICIT DOUBLE PRECISION (A-H,0-Z)
0003 COMMON /PASSW/ DUBEYU
0004 F1011=P10(Z)*P11(DUBEYU-Z)
0005 RETURN
0006 END

```

```

PDP-11 FORTRAN-77 V4.0-1 10:02:42 17-Jul-86 Page 28
SELF3M2B.FTN;16 /F77/MR

0001 DOUBLE PRECISION FUNCTION P01(Z)
0002 IMPLICIT DOUBLE PRECISION (A-H,0-Z)
0003 COMMON /ROSE/ RHOM, RHOT, GAMMA, BIGK, BIGK1
0004 P01=DEXP(RHOM*Z-RHOM)*(1.0D+RHOM*Z)
0005 RETURN
0006 END

```

```

PDP-11 FORTRAN-77 V4.0-1 10:02:44 17-Jul-86 Page 29
SELF3M2B.FTN;16 /F77/MR

0001 DOUBLE PRECISION PO1(Z)
0002 IMPLICIT DOUBLE PRECISION (A-H,0-Z)
0003 COMMON /ROSE/ RHOM, RHOT, GAMMA, BIGK, BIGK1
0004 RH1=RHOM*Z
0005 BK1=BIGK1*7
0006 D=1.0D+BK112
0007 A=BIGK*RH1Z/D
0008 PO1=BIGK*DEXP(A-RHOM)*((1.0D+A)/D)/D
0009 RETURN
0010 END

```

```

PDP-11 FORTRAN-77 V4.0-1 10:02:45 17-Jul-86 Page 30
SELF3M2B.FTN;16 /F77/MR

0001 DOUBLE PRECISION P10(Z)
0002 IMPLICIT DOUBLE PRECISION (A-H,0-Z)
0003 COMMON /ROSE/ RHOM, RHOT, GAMMA, BIGK, BIGK1
0004 RTZ=RHOT*Z
0005 BK1Z=BIGK1*Z
0006 D=BIGK-BK1Z
0007 A=RTZ/D
0008 P10=BIGK*DEXP(A-RHOT)*((1.0D+A)/D)/D
0009 RETURN
0010 END

```

PDP-11 FORTRAN-77 V4.0-1 10:02:46 17-Ju1-86
SELFL2P28.FTM:16 /F77/MR

```
0001      DOUBLE PRECISION FUNCTION P11(Z)
0002      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003      COMMON /ROSE/ RHO1, RHOT, GAMMA, BIGK, BIGK1
0004      P11=EXP(RHOT*Z/RHOT)*(1.0D+RHOT*Z)
0005      RETURN
0006      END
```

Page 31

L-12

J. S. LEE ASSOCIATES, INC.

APPENDIX M COMPUTER PROGRAM FOR PRACTICAL ADAPTIVE GAIN CONTROL RECEIVER

The following pages contain the source code listing for the FORTRAN-77 program used for numerical computations of the performance of the practical adaptive gain control receiver for FH/RMFSK.

POP-11 FORTRAN-77 V4.0-1 10:38:25 17-Jul-86
PACJAGC.FTN;7 /F77/TR:BLOCKS/MR

Page 1

POP-11 FORTRAN-77 V4.0-1 10:38:25 17-Jul-86
PACJAGC.FTN;7 /F77/TR:BLOCKS/MR

Page 2

```
0001      PROGRAM PRAC22
          C PRACTICAL ACJ/ASC RECEIVER, M=2, L=2
          C ANALYSIS: L.E. MILLER
          C PROGRAM: R.H. FRENCH
          C
          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
          PARAMETER(L=2, NSLOTS=2400)
          CHARACTER(13) FNAME
          REAL DEBNJ(126), PELOG(126)
          COMMON /ROSE/ RHOT, GAMMA, BIGK
          COMMON /PARMS/ NO, NJ, NG, DEBNOL(5), GAMBLST(10),
                         DEBNJL(126), DJ, DBJO
          S CALL ERSET(29, .TRUE., .FALSE., .TRUE., .15)
          CALL GET
          DO 900 I0=1,M0
          EBMO=10.D0**((DEBNOL(I0)/10.D0)
          RHOM=EBNO/L
          1000T=DEBNOL(10)
          DO 800 IG=1,MG
          GAMMA=GAMBLST(IG)
          10=GAMMA*NSLOTS+0.500
          16OUT=1000.D0*GAMMA+0.500
          M-2
          C PROGRESS FILE
          C
          0018      WRITE(FNAME,1) 10OUT,1GOUT
          0019      1 FORMAT('B22',12.2,14.4,'.DAT')
          0020      WRITE(5,2) FNAME
          0021      2 FORMAT(14,14,14)
          0022      OPEN(UNIT=4,FILE=FNAME,STATUS='OLD',ERR=810,
               FORM='UNFORMATTED',ACCESS='SEQUENTIAL')
          S
          C HAVE A PROGRESS FILE, READ IT
          C
          0023      READ(4) EBNOIN, GAMMIN, DBJOIN, DJIN
          IF(EBNOIN.NE.DBJOIN(10).OR. GAMMIN.NE.GAMMA .OR.
             DBJOIN.NE.DBJO .OR. DJIN.NE.DJ) STOP 'FILE CORRUPT'
          0024      S
          0025      JJ=0
          0026      801  JJ=JJ+
          0027      READ(4,END=802) DEBNJ(JJ), PELOG(JJ)
          0028      GOTO 801
          0029      802  CLOSE(UNIT=4)
          0030      GOTO 820
          C NO FILE, MUST CREATE IT
          C
          0031      810  OPEN(UNIT=4,FILE=FNAME,STATUS='NEW',FORM='UNFORMATTED',
               S
               WRITE(4) DEBNOL(10), GAMMA, DBJO, DJ
               CLOSE(UNIT=4)
          0032
          0033
```

```

POP-11 FORTRAN-77 V4.0-1 10:38:34 17-Jul-86 Page 3
PACJAGC.FTN;? /F77/TR:BLOCKS/WR

0001      SUBROUTINE GET
          C INTERACTIVE RUN PARAMETER INPUTS
          C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          DIMENSION DGAM(10), DOLST(5)
          CHARACTER*8 REPLY, BLANKS
          COMMON /PARMS/ NO, NJ, NG, DEBNL(5), DJ, DBJO
          DATA DGAM/1.0-3., 2.0-3., 5.0-3.,
          S 1.0-2., 2.0-2., 5.0-2.,
          S 1.0-1., 2.0-1., 5.0-1., 1.00/
          DATA DOLST/ 13.3524700, 12.313300, 10.9444300,
          S 14.8925300, 16.02713500/
          DATA BLANKS/, /'
          WRITE(5,2)
          2 FORMAT(' HOW MANY EB/NO? [1] ',\$)
          READ(5,3,ERR=1) NO
          3 FORMAT(1I)
          IF(NO.EQ.0) NO=1
          DO 8 IN=1,NO
          8 WRITE(5,5) IN, DOLST(IN)
          4 FORMAT(3X,'EB/NO[',1I,'] ',F8.5,' dB]: ',\$)
          5 FORMAT(3X,'EB/NO[',1I,'] ',F8.5,' dB]: ',\$)
          READ(5,6,ERR=4) REPLY
          6 FORMAT(AB)
          IF(REPLY.EQ.BLANKS) THEN
          DEBNL(IN)=DOLST(IN)
          ELSE
          READ(Reply,7,ERR=4) DEBNL(IN)
          7 FORMAT(F8.5)
          END IF
          8 CONTINUE
          9 WRITE(5,10)
          10 FORMAT(' HOW MANY GAMMA? [10] ',\$)
          READ(5,11,ERR=9) NG
          11 FORMAT(12)
          IF(NG.EQ.0) NG=10
          IF(NG.LT.0) .OR. NG.GT.10) GOTO 9
          DO 15 IN=1,NG
          12 WRITE(5,13) IN, DGAM(IN)
          13 FORMAT(3X,'GAMMA[ ',12,'] ',F5.3,']: ',\$)
          READ(5,14,ERR=12) GAMLST(IN)
          14 FORMAT(F8.6)
          IF(GAMLST(IN).EQ.0.D0) GAMLST(IN)=DGAM(IN)
          15 IF(GAMLST(IN).LE.0.D0 .OR. GAMLST(IN).GT.1.D0) GOTO 12
          16 WRITE(5,17)
          17 FORMAT(' HOW MANY EB/NJ? [126] ',\$)
          READ(5,18,ERR=16) NJ
          18 FORMAT(13)
          IF(NJ.EQ.0) NJ=126

```

```

Page 4
POP-11 FORTRAN-77 V4.0-1 10:38:34 17-Jul-86
PACJAGC.FTN;? /F77/TR:BLOCKS/WR

0046      IF(NJ.LT.0 .OR. NJ.GT.126) GOTO 16
          19 WRITE(5,20)
          20 FORMAT(' STARTING VALUE FOR EB/NJ [0 dB]: ',\$)
          21 READ(5,21,ERR=19) DBJO
          21 FORMAT(F5.2)
          DJ=0.D0
          IF(NJ.GT.1) THEN
          22 WRITE(5,23)
          23 FORMAT(' INCREMENT FOR EB/NJ [0.4 dB]: ',\$)
          24 READ(5,24,ERR=22) DJ
          24 FORMAT(F5.0)
          IF(DJ.EQ.0.00) DJ=0.400
          END IF
          RETURN
END

Page 5
POP-11 FORTRAN-77 V4.0-1 10:38:42 17-Jul-86
PACJAGC.FTN;? /F77/TR:BLOCKS/WR

0001      SUBROUTINE PSURE(1Q,PE)
          C COMPUTE ERROR PROBABILITY
          C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          PARAMETER (SLOTS=2400.00)
          PARAMETER (SLOTS1=2399.00)
          PARAMETER (SLOTPR=5757600.00)
          DIMENSION WORK(20), STACK(20), HEAP(20)
          EXTERNAL DGAU20, PDF
          COMMON /ROSE/ RHOM, RHOI, GAMMA, BIGK
          COMMON /PIES/ P10, P11, P12
          C COMPUTE ELEMENTAL EVENT PROBABILITIES
          C
          Q=10
          P10=(SLOTS-Q)*(SLOTS1-Q)/SLOTPR
          P11=Q*(SLOTS-Q)/SLOTPR
          P12=Q*(Q-1.00)/SLOTPR
          CALL ADQUA1(0.00,1.00,PE,DGAU20,PDF,1.0-4,1.0-15,
          $ IF(KODE.NE.0) STOP 'ADQUA1 ERROR'.
          G1=61.00)
          PE=2.00*PE+61*G1
          RETURN
END


```

PDP-11 FORTRAN-77 V4.0-1 10:38:44 17-Jul-86
PACJAGC.FTN;7 /F77/TR:BLOCKS/MR

Page 6

PDP-11 FORTRAN-77 V4.0-1 10:36:44 17-Jul-86
PACJAGC.FTN;7 /F77/TR:BLOCKS/MR

0001 \$
 SUBROUTINE ADQUA1(XL,XU,Y,OR,F,TOL,ABSTOL,
 WORK,STACK,HEAP,M,KODE),

C ADAPTIVE QUADRATURE ALGORITHM
C XL - LOWER LIMIT OF INTEGRAL (IN)
C XU - UPPER LIMIT OF INTEGRAL (IN)
C Y - VALUE OF INTEGRAL (OUT)
C QR - NAME OF A QUADRATURE RULE SUBROUTINE (IN)
C WITH CALLING SEQUENCE
C CALL QR(XL,XU,F,T)

C F - NAME OF FUNCTION TO BE INTEGRATED (IN)
C TOL - ERROR TOLERANCE FOR FINAL ANSWER (IN)
C ABS 'OR'-ABSOLUTE ERROR TOLERANCE (IN)
C WORK - WORK ARRAY OF SIZE N (IN)
C STACK - SECOND WORK ARRAY OF SIZE N, MUST NOT BE
C HEAP - THIRD WORK ARRAY, SIZE N, DISTINCT FROM WORK AND STACK (IN)
C N - SIZE OF WORK AND STACK; MAX. NO. OF RISECTIONS (IN)
C KODE - ERROR INDICATOR (OUT)

C 0 -- NO ERROR
C 1 -- WORK ARRAYS TOO SMALL

C R. H. FRENCH, 14 AUGUST 1984

C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
EXTERNAL F
DIMENSION WORK(N),STACK(M),HEAP(M)

KODE=0
Y=0.00
WORK(1)=XU
HEAP(1)=T

0002 \$
0003 CALL QR(XL,XU,F,T)
0004 A=XL
0005 NPTS=1
0006 EPS=TOL
0007 STACK(1)=EPS
0008 B=WORK(NPTS)
0009 10 IF=(A+B)>0.5D0
0010 CALL QR(A,XM,F,P1)
0011 CALL QR(XM,B,F,P2)
0012 TEST=MAX1(EPS*DABS(T1),ABSTOL)
0013 IF(DABS(T-P1-P2).LE.TEST) GOTO 20
0014 C SPLIT IT
0015 NPTS=NPTS+1
0016 IF(NPTS.GT.M) THEN
0017 KODE=1
0018 RETURN
0019 END IF

0020 WORK(NPTS)=XM
0021 HEAP(NPTS)=P2
0022 10 T=P1
0023 EPS=DMAX1(EPS/2,D0,5,D-16)
0024 STACK(NPTS)=EPS
0025 GOTO 10
0026

PDP-11 FORTRAN-77 V4.0-1 10:38:47 17-Jul-86
PACJAGC.FTN;7 /F77/TR:BLOCKS/MR

0031 C FINISHED A PIECE
 20 Y=Y+P1*Z
 EPS=STACK(NPTS)
 T=HEAP(NPTS)
 NPTS=NPTS-1
 A=B
 IF(NPTS.EQ.0) RETURN
 GOTO 10
 END

0032 0033
0034 0035
0035 0036
0036 0037
0037 0038
0038 C DOUBLE PRECISION FUNCTION PDF(Z)
 0001 DOUBLE PRECISION FUNCTION PDF(Z)
 0002 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
 0003 PDF=F(Z)*G(Z)
 0004 RETURN
 END

0039 0040
0040 0041
0041 0042
0042 0043
0043 0044
0044 0045
0045 0046
0046 0047
0047 0048
0048 0049
0049 0050
0050 0051
0051 0052
0052 0053
0053 0054
0054 0055
0055 0056
0056 0057
0057 0058
0058 0059
0059 0060
0060 0061
0061 0062
0062 0063
0063 0064
0064 0065
0065 0066
0066 0067
0067 0068
0068 0069
0069 0070
0070 0071
0071 0072
0072 0073
0073 0074
0074 0075
0075 0076
0076 0077
0077 0078
0078 0079
0079 0080
0080 0081
0081 0082
0082 0083
0083 0084
0084 0085
0085 0086
0086 0087
0087 0088
0088 0089
0089 0090
0090 0091
0091 0092
0092 0093
0093 0094
0094 0095
0095 0096
0096 0097
0097 0098
0098 0099
0099 0100
0100 0101
0101 0102
0102 0103
0103 0104
0104 0105
0105 0106
0106 0107
0107 0108
0108 0109
0109 0110
0110 0111
0111 0112
0112 0113
0113 0114
0114 0115
0115 0116
0116 0117
0117 0118
0118 0119
0119 0120
0120 0121
0121 0122
0122 0123
0123 0124
0124 0125
0125 0126
0126 0127
0127 0128
0128 0129
0129 0130
0130 C SPLIT IT
0130 NPTS=NPTS+1
0131 IF(NPTS.GT.M) THEN
0132 KODE=1
0133 RETURN
0134 END IF
0135 WORK(NPTS)=XM
0136 HEAP(NPTS)=P2
0137 T=P1
0138 EPS=DMAX1(EPS/2,D0,5,D-16)
0139 STACK(NPTS)=EPS
0140 GOTO 10

0039 0040
0040 0041
0041 0042
0042 0043
0043 0044
0044 0045
0045 0046
0046 0047
0047 0048
0048 0049
0049 0050
0050 0051
0051 0052
0052 0053
0053 0054
0054 0055
0055 0056
0056 0057
0057 0058
0058 0059
0059 0060
0060 0061
0061 0062
0062 0063
0063 0064
0064 0065
0065 0066
0066 0067
0067 0068
0068 0069
0069 0070
0070 0071
0071 0072
0072 0073
0073 0074
0074 0075
0075 0076
0076 0077
0077 0078
0078 0079
0079 0080
0080 0081
0081 0082
0082 0083
0083 0084
0084 0085
0085 0086
0086 0087
0087 0088
0088 0089
0089 0090
0090 0091
0091 0092
0092 0093
0093 0094
0094 0095
0095 0096
0096 0097
0097 0098
0098 0099
0099 0100
0100 0101
0101 0102
0102 0103
0103 0104
0104 0105
0105 0106
0106 0107
0107 0108
0108 0109
0109 0110
0110 0111
0111 0112
0112 0113
0113 0114
0114 0115
0115 0116
0116 0117
0117 0118
0118 0119
0119 0120
0120 0121
0121 0122
0122 0123
0123 0124
0124 0125
0125 0126
0126 0127
0127 0128
0128 0129
0129 0130
0130 C SPLIT IT
0130 NPTS=NPTS+1
0131 IF(NPTS.GT.M) THEN
0132 KODE=1
0133 RETURN
0134 END IF
0135 WORK(NPTS)=XM
0136 HEAP(NPTS)=P2
0137 T=P1
0138 EPS=DMAX1(EPS/2,D0,5,D-16)
0139 STACK(NPTS)=EPS
0140 GOTO 10

0039 0040
0040 0041
0041 0042
0042 0043
0043 0044
0044 0045
0045 0046
0046 0047
0047 0048
0048 0049
0049 0050
0050 0051
0051 0052
0052 0053
0053 0054
0054 0055
0055 0056
0056 0057
0057 0058
0058 0059
0059 0060
0060 0061
0061 0062
0062 0063
0063 0064
0064 0065
0065 0066
0066 0067
0067 0068
0068 0069
0069 0070
0070 0071
0071 0072
0072 0073
0073 0074
0074 0075
0075 0076
0076 0077
0077 0078
0078 0079
0079 0080
0080 0081
0081 0082
0082 0083
0083 0084
0084 0085
0085 0086
0086 0087
0087 0088
0088 0089
0089 0090
0090 0091
0091 0092
0092 0093
0093 0094
0094 0095
0095 0096
0096 0097
0097 0098
0098 0099
0099 0100
0100 0101
0101 0102
0102 0103
0103 0104
0104 0105
0105 0106
0106 0107
0107 0108
0108 0109
0109 0110
0110 0111
0111 0112
0112 0113
0113 0114
0114 0115
0115 0116
0116 0117
0117 0118
0118 0119
0119 0120
0120 0121
0121 0122
0122 0123
0123 0124
0124 0125
0125 0126
0126 0127
0127 0128
0128 0129
0129 0130
0130 C SPLIT IT
0130 NPTS=NPTS+1
0131 IF(NPTS.GT.M) THEN
0132 KODE=1
0133 RETURN
0134 END IF
0135 WORK(NPTS)=XM
0136 HEAP(NPTS)=P2
0137 T=P1
0138 EPS=DMAX1(EPS/2,D0,5,D-16)
0139 STACK(NPTS)=EPS
0140 GOTO 10

0039 0040
0040 0041
0041 0042
0042 0043
0043 0044
0044 0045
0045 0046
0046 0047
0047 0048
0048 0049
0049 0050
0050 0051
0051 0052
0052 0053
0053 0054
0054 0055
0055 0056
0056 0057
0057 0058
0058 0059
0059 0060
0060 0061
0061 0062
0062 0063
0063 0064
0064 0065
0065 0066
0066 0067
0067 0068
0068 0069
0069 0070
0070 0071
0071 0072
0072 0073
0073 0074
0074 0075
0075 0076
0076 0077
0077 0078
0078 0079
0079 0080
0080 0081
0081 0082
0082 0083
0083 0084
0084 0085
0085 0086
0086 0087
0087 0088
0088 0089
0089 0090
0090 0091
0091 0092
0092 0093
0093 0094
0094 0095
0095 0096
0096 0097
0097 0098
0098 0099
0099 0100
0100 0101
0101 0102
0102 0103
0103 0104
0104 0105
0105 0106
0106 0107
0107 0108
0108 0109
0109 0110
0110 0111
0111 0112
0112 0113
0113 0114
0114 0115
0115 0116
0116 0117
0117 0118
0118 0119
0119 0120
0120 0121
0121 0122
0122 0123
0123 0124
0124 0125
0125 0126
0126 0127
0127 0128
0128 0129
0129 0130
0130 C SPLIT IT
0130 NPTS=NPTS+1
0131 IF(NPTS.GT.M) THEN
0132 KODE=1
0133 RETURN
0134 END IF
0135 WORK(NPTS)=XM
0136 HEAP(NPTS)=P2
0137 T=P1
0138 EPS=DMAX1(EPS/2,D0,5,D-16)
0139 STACK(NPTS)=EPS
0140 GOTO 10

0039 0040
0040 0041
0041 0042
0042 0043
0043 0044
0044 0045
0045 0046
0046 0047
0047 0048
0048 0049
0049 0050
0050 0051
0051 0052
0052 0053
0053 0054
0054 0055
0055 0056
0056 0057
0057 0058
0058 0059
0059 0060
0060 0061
0061 0062
0062 0063
0063 0064
0064 0065
0065 0066
0066 0067
0067 0068
0068 0069
0069 0070
0070 0071
0071 0072
0072 0073
0073 0074
0074 0075
0075 0076
0076 0077
0077 0078
0078 0079
0079 0080
0080 0081
0081 0082
0082 0083
0083 0084
0084 0085
0085 0086
0086 0087
0087 0088
0088 0089
0089 0090
0090 0091
0091 0092
0092 0093
0093 0094
0094 0095
0095 0096
0096 0097
0097 0098
0098 0099
0099 0100
0100 0101
0101 0102
0102 0103
0103 0104
0104 0105
0105 0106
0106 0107
0107 0108
0108 0109
0109 0110
0110 0111
0111 0112
0112 0113
0113 0114
0114 0115
0115 0116
0116 0117
0117 0118
0118 0119
0119 0120
0120 0121
0121 0122
0122 0123
0123 0124
0124 0125
0125 0126
0126 0127
0127 0128
0128 0129
0129 0130
0130 C SPLIT IT
0130 NPTS=NPTS+1
0131 IF(NPTS.GT.M) THEN
0132 KODE=1
0133 RETURN
0134 END IF
0135 WORK(NPTS)=XM
0136 HEAP(NPTS)=P2
0137 T=P1
0138 EPS=DMAX1(EPS/2,D0,5,D-16)
0139 STACK(NPTS)=EPS
0140 GOTO 10

0039 0040
0040 0041
0041 0042
0042 0043
0043 0044
0044 0045
0045 0046
0046 0047
0047 0048
0048 0049
0049 0050
0050 0051
0051 0052
0052 0053
0053 0054
0054 0055
0055 0056
0056 0057
0057 0058
0058 0059
0059 0060
0060 0061
0061 0062
0062 0063
0063 0064
0064 0065
0065 0066
0066 0067
0067 0068
0068 0069
0069 0070
0070 0071
0071 0072
0072 0073
0073 0074
0074 0075
0075 0076
0076 0077
0077 0078
0078 0079
0079 0080
0080 0081
0081 0082
0082 0083
0083 0084
0084 0085
0085 0086
0086 0087
0087 0088
0088 0089
0089 0090
0090 0091
0091 0092
0092 0093
0093 0094
0094 0095
0095 0096
0096 0097
0097 0098
0098 0099
0099 0100
0100 0101
0101 0102
0102 0103
0103 0104
0104 0105
0105 0106
0106 0107
0107 0108
0108 0109
0109 0110
0110 0111
0111 0112
0112 0113
0113 0114
0114 0115
0115 0116
0116 0117
0117 0118
0118 0119
0119 0120
0120 0121
0121 0122
0122 0123
0123 0124
0124 0125
0125 0126
0126 0127
0127 0128
0128 0129
0129 0130
0130 C SPLIT IT
0130 NPTS=NPTS+1
0131 IF(NPTS.GT.M) THEN
0132 KODE=1
0133 RETURN
0134 END IF
0135 WORK(NPTS)=XM
0136 HEAP(NPTS)=P2
0137 T=P1
0138 EPS=DMAX1(EPS/2,D0,5,D-16)
0139 STACK(NPTS)=EPS
0140 GOTO 10

0039 0040
0040 0041
0041 0042
0042 0043
0043 0044
0044 0045
0045 0046
0046 0047
0047 0048
0048 0049
0049 0050
0050 0051
0051 0052
0052 0053
0053 0054
0054 0055
0055 0056
0056 0057
0057 0058
0058 0059
0059 0060
0060 0061
0061 0062
0062 0063
0063 0064
0064 0065
0065 0066
0066 0067
0067 0068
0068 0069
0069 0070
0070 0071
0071 0072
0072 0073
0073 0074
0074 0075
0075 0076
0076 0077
0077 0078
0078 0079
0079 0080
0080 0081
0081 0082
0082 0083
0083 0084
0084 0085
0085 0086
0086 0087
0087 0088
0088 0089
0089 0090
0090 0091
0

POP-11 FORTRAN-77 V4.0-1 10:38:51 17-Jul-86
 Page 10
 PACAGC.FTN;? /F77/TR-BLOCKS/MR

 0001 DOUBLE PRECISION FUNCTION G(X)
 0002 IMPLICIT DOUBLE PRECISION(A-H,O-Z)
 0003 COMMON /PIES/ P10, P11, P12
 0004 COMMON /ROSE/ RHOM, RHOT, GAMMA, BIGK
 0005 X1=X+1. DO
 0006 PART1=DEXP(-RHOM/X1)/X1
 0007 PART2=DEXP(-RHOT/X1)/X1
 0008 XK1=1.00+BIGK*X
 0009 PART3=BIGK*DEXP(-RHOM/XK1)/XK1
 0010 XK=X+BIGK
 0011 PART4=DEXP(-BIGK*RHOT/XK)/XK
 0012 G=X*(P10*PART1+P11*(PART4+PART3)+P12*PART2)
 0013 RETURN
 0014 END

J. S. LEE ASSOCIATES, INC.

REFERENCES

1. J.S. Lee, L.E. Miller, R.H. French, Y.K. Kim, and A.P. Kadrichu, "The optimum jamming effects on frequency-hopping M-ary FSK systems under certain ECCM receiver design strategies, Final report, Office of Naval Research Contract N00014-83-C-0312, October 1984. (AD-A147766)
2. Milton Abramowitz and Irene A. Stegun (eds.), Handbook of Mathematical Functions, National Bureau of Standards Applied Mathematics Series 55. Washington, D.C.: Government Printing Office, June 1964, Ninth printing, November 1970.
3. I.S. Gradshteyn and I.M. Ryzhik, Table of Integrals, Series, and Products, Corrected and Enlarged Edition. New York: Academic Press, 1980.
4. Daniel I.A. Cohen, Basic Techniques of Combinatorial Theory. New York: Wiley and Sons, 1978.
5. A.J. Viterbi and I.M. Jacobs, "Advances in Coding and Modulation for Non-Coherent Channels Affected by Fading, Partial Band, and Multiple Access Interference", in Advances in Comm. Syst., A.J. Viterbi (ed.), New York: Academic Press, 1975, pp. 279-308.
6. J.E. Blanchard, "A Slow Frequency Hopping Technique That is Robust to Repeat Jamming," IEEE Military Electronics Conf. Record, October 1982, paper 14.1.
7. D.J. Torrieri, Principles of Military Communications. Dedham, MA: Artech, 1981.
8. H. Hite, et al., "High Frequency Counter-Countermeasures Study", August 5, 1983, Hughes Aircraft Co., Fullerton, CA, under Contract DAAK80-81-C-0119, AD-C032993.
9. D.J. Torrieri, "Information-Bit Error Rate for Block Codes", IEEE Trans. on Comm., Vol. COM-32, April 1984, pp. 474-476.
10. J.S. Lee, et al., "Probability of Error Analyses of a BFSK Frequency Hopping System With Diversity Under Partial-Band Jamming Interference - Part I: Performance of Square-Law Linear Combining Soft Decision Receiver", IEEE Trans. on Commun., Vol. COM-32, June 1984, pp. 645-653.
11. J.S. Lee, et al., "Probability of Error Analyses of a BFSK Frequency Hopping System With Diversity Under Partial-Band Jamming Interference - Part II: Performance of Square-Law Nonlinear Combining Soft Decision Receivers", IEEE Trans. on Commun., Vol. COM-32, December 1984, pp. 1243-1250.
12. K.S. Gong, "Performance of Diversity Combining Techniques for FH/MFSK in Worst Case Partial Band Noise and Multi-Tone Jamming", Proc. 1983 IEEE Military Commun. Conf., pp. 17-21.

J. S. LEE ASSOCIATES, INC.

13. Y.K. Kim, et al., "The Exact Performance Analyses of Two Types of Adaptive Receivers for Multi-Hops Per Symbol FH/MFSK Systems in Partial-Band Noise Jamming and System Thermal Noise", Proc. IEEE Global Telecom. Conf., 1983, pp. 1309-1314.
14. J.S. Lee, et al., "Signal Design and Detection Strategies for LPI Communications in Electronic Warfare Environments", J.S. Lee Associates, Inc., May 1983, Final report on Contract N00014-81-C-0534, AD-B073026L.
15. J.S. Lee, et al., "Error Performance Analyses of Linear and Nonlinear Combining Square-Law Receivers for L-Hops Per Bit FH/BFSK Waveforms in Worst-Case Partial-Band Jamming", Proc. 1983 IEEE Military Comm. Conf., pp. 22-28.
16. J.S. Bird and E.B. Felstead, "Antijam Performance of Fast Frequency-Hopped M-ary NCFSK an Overview," IEEE Journal on Selected Areas in Communications, Vol. SAC-4, March 1986, pp. 216-233.
17. D.J. Torrieri, "Frequency Hopping with Multiple Frequency-Shift Keying and Hard Decisions," IEEE Trans. on Commun., Vol. COM-32, May 1984, pp. 574-582.
18. E.J. Kelly, et al., "The Sensitivity of Radiometric Measurements", J. Soc. Indust. Appl. Math., Vol. 11, pp. 235-257, June 1963.
19. J.S. Bendat and A.G. Piersol, Random Data: Analysis and Measurement Procedures, New York: Wiley-Interscience, 1971.
20. A. Gelb, ed., Applied Optimal Estimation. Cambridge: MIT Press, 1974.
21. L.E. Miller, J.S. Lee and A.P. Kadrichu, "Probability of Error Analyses of a BFSK Frequency-Hopping System with Diversity under Partial-Band Jamming Interference--Part III: Performance of a Square-Law Self-Normalizing Soft Decision Receiver," IEEE Trans. on Commun., Vol. COM-34, No. 7, July 1986, pp. 669-675.